Groupement d'Intéret Scientifique
Surveillance, Sureté et Sécurité des Grands Systèmes
– GIS 3SGS –

Projet ACDA-P2P :
Approche Collaborative pour la Détection d'Attaques
dans les réseaux Pair à Pair

Délivrable 2

Vulnérabilités de la DHT de BitTorrent & Identification des comportements malveillants dans KAD

UMR STMR 6279 :
Thibault Cholez
Guillaume Doyen
Rida Khatoun

LORIA–INRIA Nancy Grand Est :
Juan Pablo Timpanaro
Isabelle Chrisment
Olivier Festor

23 novembre 2011

Résumé

Le présent délivrable présente les résultats des travaux menés durant les six premiers mois (T0+6) du projet GIS 3SGS Acda-P2P dont l'objectif est de proposer une architecture collaborative pour la détection d'attaques dans les réseaux pair à pair. Nous détaillons dans ce rapport nos travaux concernant l'identification des comportements malveillants affectant le réseaux KAD (tâche T2) ainsi que l'identification des vulnérabilités affectant la DHT du réseau BitTorrent (tâche T3) qui sont au coeur du projet Acda-P2P.

Pour introduire nos travaux, nous présentons tout d'abord leur contexte ainsi qu'une taxonomie des différentes attaques pouvant affecter les DHT. Nous différencions notamment les attaques internes à la DHT, que nous traitons dans ce rapport, des attaques externes qui sont utilisées notamment pour diffuser la pollution et qui feront l'objet du prochain rapport.

Notre première contribution consiste en l'étude de la sécurité de la DHT du réseau BitTorrent (Mainline DHT) qui est de plus en plus utilisée afin de remplacer les serveurs « trackers » par un système complètement distribué. Nous montrons à travers plusieurs expériences que des failles de sécurité permettent la réalisation d'attaques efficaces pouvant altérer le bon fonctionnement du réseau. Nous montrons également l'applicabilité d'une solution issue de nos précédents travaux à la DHT de BitTorrent, et qui permettrait de limiter significativement ces attaques.

En prenant pour cas d'étude le réseau P2P KAD, nous recensons ensuite les pairs suspects en utilisant deux approches de détection, l'une basée sur l'analyse des distances inter-pairs et l'autre sur l'analyse des distances pairs-contenus. Nous analysons pour cela une cartographie du réseau obtenue grâce à l'implantation d'un explorateur permettant de découvrir l'ensemble des pairs avec une grande précision. Nous montrons ainsi que des milliers de contenus du réseau sont attaqués durant nos mesures.

Finalement, nous avons souhaité caractériser les attaquants ciblant les contenus populaires. Nous avons réalisé pour cela des observations du réseau P2P que nous avons corrélé à une base de données de contenus multimédia permettant d'obtenir leur popularité dans le temps. Nous expliquons en quoi l'absence d'attaques relevées durant cette seconde campagne de mesure rend l'exploitation de ces données impossible, ce qui a motivé l'évolution de la suite du projet vers la détection des attaques externes propageant la pollution.

Table des matières

Chapitre 1

Introduction

Le projet Acda-P2P est un projet académique, financé par le GIS [1] 3SGS [2]. Il regroupe l'équipe ERA [3] de l'UMR 6279 STMR [4] et l'équipe MADYNES [5] de l'INRIA [6] Grand Est. D'une durée initiale d'un an, le projet a débuté en mai 2010. Il vise à proposer une solution collaborative pour la détection d'attaques dans les réseaux pair à pair (P2P). Plus spécifiquement, il s'inscrit dans un contexte scientifique détaillé dans le premier délivrable et résumé ici.

Les réseaux pair à pair (P2P), notamment ceux utilisant les tables de hachage distribuées, sont devenus, en quelques années, une application majeure de l'Internet en permettant à des millions d'utilisateurs de partager rapidement et sans coût d'infrastructure de grandes quantités de données. Cependant, les réseaux P2P peuvent également être un support pour des activités malveillantes menaçant la sécurité du réseau P2P lui-même (pollution des données, surveillance des échanges, ...) ou, plus généralement, d'Internet (déni de service, propagation de vers, contrôle de botnet, ...). Étant donné le développement croissant de ces réseaux, pouvoir détecter lorsqu'un réseau P2P devient le support d'activités malveillantes devient primordial pour s'en prémunir.

Le premier délivrable a présenté les différentes architectures P2P parmi lesquelles les Table de Hachage Distribuées (DHT), qui ont été retenues pour notre étude de part leurs qualités intrinsèques et leur déploiement à grande échelle. Celles-ci souffrent néanmoins de nombreuses attaques que nous avons précédemment décrites. En particulier, la vulnérabilité la plus critique consiste en l'insertion ciblée de nœuds malveillants pouvant prendre le contrôle des références stockées au sein du réseau. Ces réseaux étant entièrement dynamiques et distribués, il est très difficile de collecter des informations afin de détecter les attaques, et encore davantage d'agir sur des comportements malveillants au sein du réseau. Les précédentes approches de supervision dans les réseaux P2P ont totalement omis les considérations de sécurité. Elles s'intéressent majoritairement à obtenir des données synthétiques (trafic généré, données topologiques, comportement général des pairs) sur les réseaux P2P, parfois sur leurs contenus, mais sans jamais s'intéresser aux motifs traduisant une utilisation anormale de la DHT.

1. Groupement d'Intérêt Scientifique
2. Surveillance, Sureté et Sécurité des Grands Systèmes
3. Environnement de Réseaux Autonomes
4. Science et Technologie pour la Maîtrise des Risques
5. Management of Dynamic Networks and Services
6. Institut National de Recherche en Informatique et Automatique

Parmi les méthodes de supervision étudiées dans le précédent délivrable, l'approche passive permet d'observer sur une petite zone de la DHT le trafic P2P sans injecter de données supplémentaires dans le réseau. Les honeypots permettent également de collecter des informations sur l'activité de quelques contenus spécifiques au sein du réseau P2P, en attirant les pairs malveillants par l'annonce de faux fichiers. Ces approches offrent une vision précise mais cependant trop localisée et fragmentaire du réseau global. Les méthodes de supervision actives sollicitant quant à elle directement les pairs et sont souvent trop intrusives et détectables. Les explorateurs, peuvent découvrir l'ensemble des pairs d'un réseau mais ne peuvent appréhender les communications (et services) échangés entre pairs. Une solution collaborative pourrait permettre de combiner les qualités des outils de détection localisés tout en étant capable de considérer le réseau dans son intégralité.

Dans ce contexte, ce délivrable présente les résultats des tâches T2 et T3 du projet ACDA-P2P, à savoir, l'étude des vulnérabilités de la DHT de BitTorrent et l'identification des comportements malveillants dans les réseaux P2P. Nous proposons tout d'abord une taxonomie des principales attaques (surveillance des échanges, pollution des contenus, etc.) de la DHT en considérant les deux principaux modes opératoires possibles, à savoir les attaques internes à la DHT et les attaques externes.

Nous nous intéressons ensuite à l'étude des vulnérabilités de la DHT du réseau BitTorrent ainsi que de l'applicabilité d'un mécanisme de protection contre l'attaque Sybil dans ce même réseau (T3). Ces travaux ont été réalisés exclusivement par le LORIA qui en a validé les résultats par une publication internationale [TCCF11] [7]. Bien que BitTorrent ait été montré comme vulnérables aux attaques, nous avons retenu KAD comme réseau cible pour les travaux suivants car les services offerts par sa DHT sont plus riches et attirent davantage de pairs malveillants souhaitant les corrompre. Par conséquent, BitTorrent ne sera pas utilisé comme support des expérimentations réalisées dans le cadre du projet ACDAP2P.

Dans un second temps, nous nous intéressons à la détection, par une approche centralisée, des attaques ciblées dans un autre réseau P2P largement déployé, à savoir KAD (T2). Nous avons conçu pour cela un explorateur, dont nous détaillons le fonctionnement, qui nous permet d'obtenir une vue précise du réseau. Nous analysons ensuite les données collectées en considérant d'une part la densité locale des pairs dans la DHT et d'autre part la distance entre les pairs et certains contenus populaires, ce qui nous permet de découvrir de nombreux pairs déviants. Ces travaux ont été réalisés en collaboration entre le LORIA et l'UTT et ont été validés par une publication nationale [CHC+11] [8].

Enfin, nous avons souhaité étudier plus précisément le comportement de ces pairs suspects par des mesures quotidiennes autour de mots-clés, choisis après consultation d'une base de données de contenus multimédia. L'analyse des données collectées, par des méthodes de statistiques descriptives et inférentielles, devait permettre de caractériser en partie le comportement de ces pairs déviants. Cependant, la très faible quantité d'attaques relevées durant la période de mesure ne permet pas une telle exploitation des données.

Les contributions sont organisées comme suit : le chapitre 2 présente une taxonomie des principales attaques des DHT. Le chapitre 3 présente l'étude des vulnérabilités de la DHT de BitTorrent ainsi qu'une solution possible de protection. Le chapitre 4 présente ensuite la détection de nœuds suspects après exploration du réseau KAD. Nous analysons dans le chapitre

7. http://hal.inria.fr/inria-00577043/PDF/BitTorrent_DHT_security_assessment_ntms11.pdf
8. http://hal.inria.fr/inria-00596677/PDF/SARSSI11-Detection_Attaques_KAD-Cholez.pdf

5 des relevés quotidiens de pairs à proximité de contenus multimédia. Enfin, le chapitre 6 conclut et indique la suite des travaux menés dans le projet ACDA-P2P. Ces derniers consistent principalement en l'étude du second problème majeur de sécurité affectant les réseaux P2P et mis en évidence par la taxonomie, à savoir la pollution du réseau (T4), puis en la conception de sondes autonomes capables détecter les contenus pollués (T5).

Chapitre 2

Taxonomie des attaques sur la DHT

2.1 Introduction

Les réseaux pair à pair (P2P) sont devenus une application majeure d'Internet en permettant à leurs utilisateurs de partager rapidement et sans coût d'infrastructure de grandes quantités de données. Parmi les différentes architectures pair à pair complètement distribuées, les tables de hachage distribuées (DHT) ont prouvé aussi bien en théorie qu'en pratique leur capacité à constituer des systèmes d'information performants. Basés sur l'architecture Kademlia, les réseaux P2P tels que KAD ou la DHT de BitTorrent (Mainline DHT) regroupent ainsi des millions d'utilisateurs.

Si l'absence de composant central apporte au paradigme P2P ses principaux avantages (passage à l'échelle, robustesse, absence de coûts d'infrastructure), il constitue également une limite en rendant difficile l'application de règles de sécurité. En effet, les pairs étant parfaitement autonomes, certains pairs malveillants peuvent détourner le protocole à leurs propres fins, telles que : la surveillance des échanges [SENB07a] [MRGS09] [CCF10b], la pollution [LNR06] [LMSW10] [CCF10b], la suppression d'information [SENB07a] [KLR09] ou encore le déni de service distribué [NR06] [SENB07a] [WTCT$^+$08]. Si plusieurs attaques pouvant affecter les tables de hachage distribuées sont d'ores et déjà connues (attaque Sybil, attaque ciblée) et ont été expérimentées ponctuellement, aucune étude à ce jour ne s'est intéressée à recenser de telles attaques en pratique.

2.2 Contexte et travaux relatifs

2.2.1 Le réseau KAD

KAD est un réseau P2P structuré basé sur le protocole de routage Kademlia [MM02] et implanté par les clients libres eMule [1] et aMule [2] qui permettent le partage de fichiers entre utilisateurs. Rendu populaire au fil des fermetures des serveurs eDonkey, KAD est principalement utilisé en Europe et en Chine et compte environ 3 millions d'utilisateurs simultanés, ce qui en fait l'un des plus importants réseaux P2P déployés.

1. http ://www.emule-project.net/
2. http ://www.amule.org/

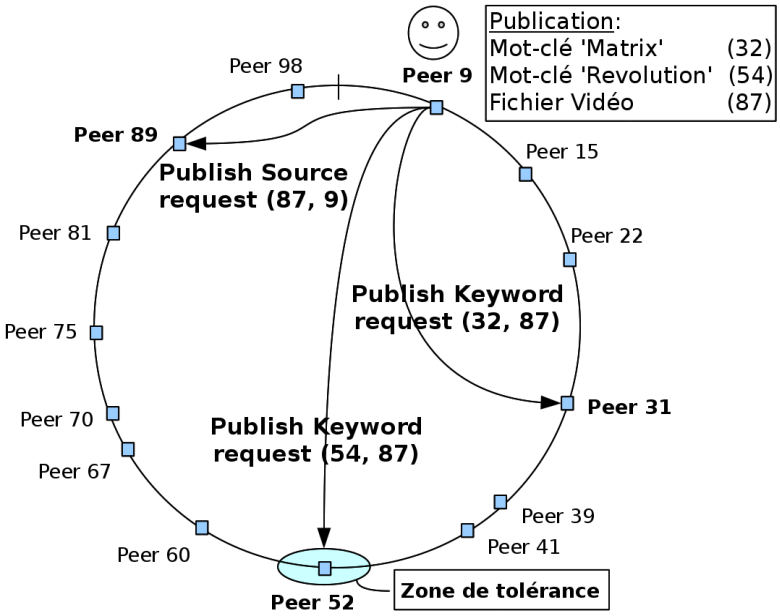

FIGURE 2.1 – Indexation à deux niveau sur KAD

Dans KAD, chaque pair ainsi que chaque information indexée dans le réseau possède un identifiant « KADID » de 128 bits définissant sa place sur la DHT. Le routage est basé sur la métrique XOR grâce à laquelle on mesure la distance entre deux identifiants. La table de routage de chaque pair est organisée en un arbre dont les feuilles sont constituées de groupes de taille constante de K contacts ($K = 10$), la distance entre les contacts retenus et le pair courant étant divisée par deux (1 bit supplémentaire commun) à chaque niveau de l'arbre. Ainsi le niveau **i** représente une portion du réseau de taille $n/2^i$, donc d'autant plus petite que celle-ci est proche du pair courant. Cette organisation permet de localiser efficacement les identifiants recherchés en $O(log\ n)$ messages, n étant la taille du réseau.

En tant que support au partage de fichiers, la fonction principale de la DHT de KAD est d'indexer des mots-clés et des fichiers selon la procédure présentée par la figure 2.1. Lorsqu'un fichier est partagé, dans l'exemple « matrix_revolution.avi », son contenu ainsi que chaque mot-clé constituant le nom du fichier sont hachés par une fonction MD4 (donnant les identifiants 32, 54, 87 pour respectivement chacun des mots-clés et le fichier). Les identifiants ainsi générés sont ensuite publiés sur le réseau. Les pairs chargés de l'indexation d'une information sont les dix pairs dont les identifiants sont les plus proches de celui de l'information.

Un mécanisme de double indexation permet de retrouver un fichier correspondant à un ensemble de mots-clés. Pour publier un fichier, deux types de requêtes sont nécessaires :

- les requêtes *KADEMLIA2_PUBLISH_KEY_REQ* sont envoyées vers l'identifiant des mots-clés et associent un mot-clé (32 ou 54) avec un fichier (87) ;
- les requêtes *KADEMLIA2_PUBLISH_SOURCE_REQ* sont envoyées vers l'identifiant du fichier (87) et associent un fichier avec une source (le pair 9 le partageant).

La réalisation de services (publication ou recherche) se fait en deux étapes. Dans un premier temps, le processus de localisation trouve les pairs les plus proches de l'identifiant de l'information visée (en émettant des requêtes *KADEMLIA2_REQ* de manière itérative), puis les requêtes

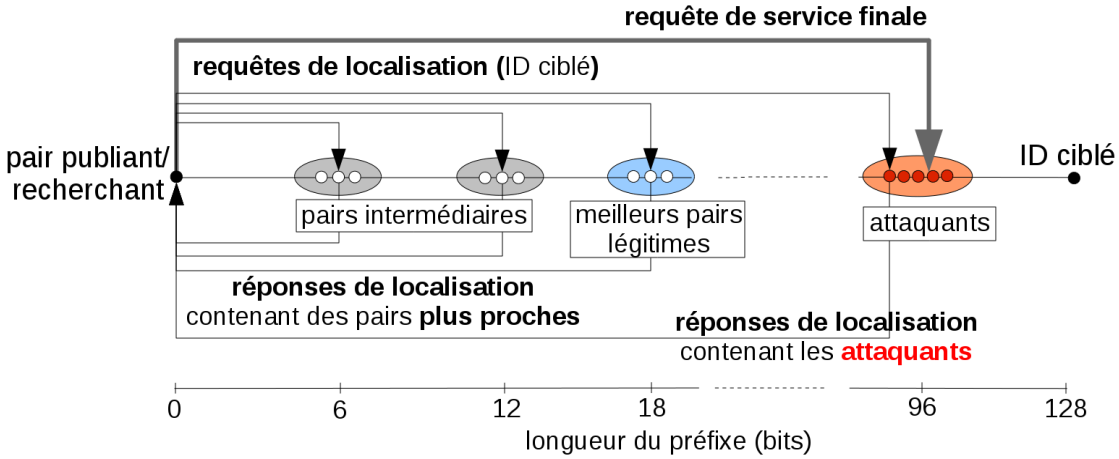

FIGURE 2.2 – Prise de contrôle d'une référence sur la DHT de KAD

spécifiques au service demandé sont envoyées à ces pairs.

2.2.2 La sécurité des DHT

Plusieurs problèmes de sécurité ont été mis en évidence dans cette architecture. En réalisant une attaque Sybil [Dou02] dans KAD qui consiste à insérer de nombreux pairs factices (appelés « Sybils ») contrôlés par une même entité, les auteurs de [SENB07a] ont montré que le réseau était très vulnérable et pouvait être largement affecté par une attaque émise d'une seule machine. En effet, après avoir découvert les pairs d'une zone de la DHT, les auteurs ont pu y injecter de nombreux Sybils ($2^{16} = 65535$ contre environ 10000 pairs légitimes) obtenant ainsi le contrôle de cette zone en y interceptant la grande majorité des messages. En restreignant la zone d'attaque au voisinage immédiat d'une information, il est également possible d'en prendre le contrôle avec moins de Sybils (une vingtaine). Le vecteur d'attaque utilisé ici est la table de routage des pairs, les Sybils s'annonçant directement pour se propager.

Dès lors, certains mécanismes de protection ont été implantés pour protéger la table de routage de telles attaques [CCF09]. De nouvelles contraintes empêchent dorénavant deux pairs affichant une même adresse IP d'être insérés dans une même table de routage. De même, deux pairs appartenant au même sous-réseau ne peuvent pas être trop proches dans une même table de routage, c'est à dire dans la même feuille de l'arbre. Cependant, nous avons montré dans nos précédents travaux [CCF10b] que les attaques ciblées peuvent utiliser des nœuds distribués sur le réseau IP et continuer d'être efficaces avec peu de ressources. Le schéma 2.2 montre les échanges de messages nécessaires à la réalisation d'un service sur KAD lorsqu'une référence est attaquée. Les pairs malveillants sont ainsi insérés plus proches que n'importe quels autres de la ressource visée (96 bits en commun) et coopèrent pour attirer les requêtes de service.

Plusieurs applications exploitent cette vulnérabilité. Les nœuds ainsi insérés en des points spécifiques constituent autant de sondes capables de surveiller les messages échangés au sein du réseau P2P. [SENB07a] surveille ainsi une portion complète de la DHT, [CCF10b] s'intéresse à des mots-clés spécifiques et annonce des pots de miel alors que [MRGS09] place des sondes de manière à recevoir une copie du trafic émis vers chaque pair du réseau. Ces pratiques posent

des problèmes de vie privée pour les utilisateurs du réseau.

D'autres attaques sont également possibles : les auteurs de [SENB07a] ont réalisé une attaque de type éclipse faisant disparaître de l'index du réseau le contenu ciblé. Ils ont aussi expérimenté, tout comme [NR06], un déni de service distribué en injectant systématiquement l'adresse IP d'une machine victime dans les réponses émises par les Sybils et générant ainsi plus de 100Mbit/sec de trafic. Les auteurs de [LNR06] ont montré que la DHT d'Overnet pouvait être polluée efficacement par l'insertion de nœuds autour de certains mots-clés. Ce problème affecte également KAD [LMSW10]. Nous avons montré dans [CCF10b] que l'attaque locale permettait en outre de polluer efficacement le réseau en générant à faible coût de faux fichiers très attractifs ce qui peut amener les utilisateurs à télécharger des contenus indésirables et illégaux (virus, contenu pédophile, ...) à leur insu.

Bien que nous ayons proposé dans [CCF10a] une méthode capable de détecter les attaques ciblées analysant la distribution des identifiants autour d'une ressource sur la DHT, celle-ci n'est pas déployée à grande échelle, laissant le réseau vulnérable aux attaques sus-mentionnées.

2.3 Taxonomie

Etant donné l'état de l'art, nous proposons de classifier les problèmes de sécurité des DHTs en deux groupes selon que l'insertion des pairs malveillant au sein de la DHT est un élément de l'attaque ou non.

2.3.1 Attaques internes à la DHT

Les attaques internes à la DHT reposent sur l'insertion de pairs malveillants en son sein et contrôlés par une même entité (Sybil attaque). Les pairs insérés peuvent être passifs, c'est à dire participer normalement à la DHT mais en enregistrant les messages reçus ce qui permet une supervision du réseau [MRGS09] [CCF10b], ou actifs, et forgent alors les réponses à des fins diverses (déni de service, pollution, éclipse, etc.). Le second paramètre permettant de distinguer entre les attaques internes concernent leur étendue : celles-ci peuvent être limitées à quelques contenus particuliers et donc localisées autour des identifiants de ceux-ci sur la DHT [MRGS09] [CCF10b] [KLR09] ou au contraire concerner une zone étendue de la DHT [SENB07a] [CCF10b], indifféremment des contenus stockés dans celle-ci. Le dernier paramètre concerne la distribution de l'attaque : les noeuds insérés peuvent provenir d'une même machine [SENB07a] ou au contraire de machines distribuées sur le réseau Internet qui rend les contre-mesures beaucoup plus difficiles [CCF10b].

2.3.2 Attaques externes à la DHT

La grande majorité des attaques contre les DHT étudiées dans la littérature sont des attaques internes. Cependant, le réseau peut également être perturbé par des attaquants sans que ceux-ci y participent. Ainsi, les attaques externes reposent sur le fait de découvrir les pairs du réseau, en utilisant un explorateur, puis d'émettre des messages corrompus vers ces derniers pour réaliser l'attaque. La pollution du réseau [LMSW10] peut notamment être réalisée de cette façon en annonçant de fausses références aux pairs responsables d'un mot-clé. Par définition, une attaque

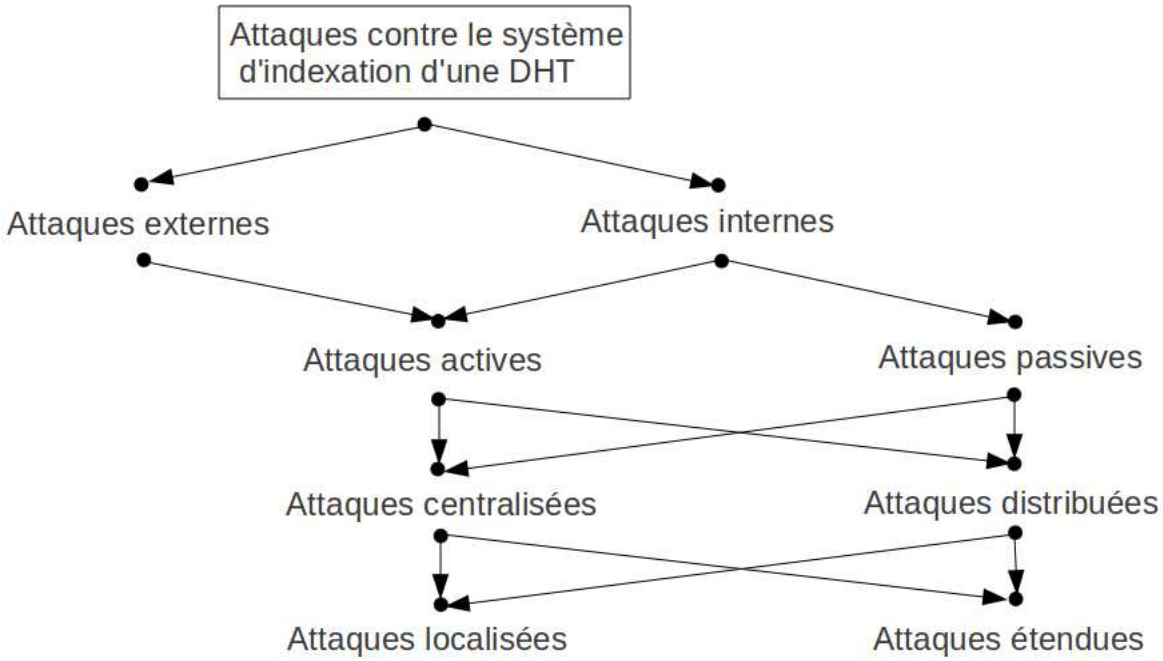

FIGURE 2.3 – Taxonomie des attaques pouvant affecter une DHT

externe ne peut être passive puisque des messages doivent être émis par les attaquants. Une attaque externe peut en revanche être localisée ou étendue, et centralisée ou distribuée, tout comme une attaque interne. Le schéma 2.3 résume notre classification des différentes attaques affectant les DHT.

2.4 Conclusion

Nous avons présenté dans ce chapitre le fonctionnement d'une DHT largement déployée, à savoir KAD, et les nombreux problèmes de sécurité qui ont été mis en évidence dans la littérature pour ce type d'architecture. Nous avons en outre proposé une classification des attaques selon leur méthode de mise en oeuvre. Nous nous focalisons dans les prochains chapitres sur l'étude des attaques internes car celles-ci semblent plus néfastes de part le contrôle de la DHT qu'elles permettent et la littérature conséquente à leur sujet. Dans le chapitre 3, nous montrons tout d'abord que le système de DHT récemment mis en place par BitTorrent est lui aussi vulnérable aux attaques internes et rappelons l'existence d'une solution possible contre celles-ci. Dans le chapitre 4, nous nous intéressons pour la première fois à la métrologie des attaques internes réellement lancées contre un réseau P2P réel dans le cas de KAD.

Chapitre 3

Identification des vulnérabilités de la DHT de BitTorrent

3.1 Introduction

BitTorrent [Coh03] est un protocole P2P de partage de fichiers très populaire développé par Bram Cohen. Il est notamment utilisé pour la distribution de contenus multimédia et de mises à jour de logiciels. Une étude récente [Ipo09] a montré que, selon les zones géographiques considérées, BitTorrent représente entre 43% et 70% du trafic d'Internet ce qui en fait de loin le protocole P2P le plus utilisé. Cependant, des actions légales à l'encontre des serveurs offrant des services de découverte de fichiers « torrents » ou de pairs « serveur tracker » ont récemment été menées par des entreprises produisant des biens culturels (RIAA [1]) et mettent en péril la pérennité du réseau. Par ailleurs, certains pays ont relayé ces actions à travers des procédures de filtrage du trafic visant à bloquer l'accès à des sites d'indexation tels que the Pirate Bay ou Mininova, indépendamment des contenus indexés. Sans ces sites d'indexation centralisés permettant de trouver les contenus et les pairs associés, le réseau n'est plus utilisable. Ces attaques ont motivé une évolution du protocole vers une solution complètement distribuée reposant sur l'utilisation d'une table de hachage distribuée (DHT) capable d'assurer le service de recherche de torrents et des pairs associés. Ainsi, chaque pair agit comme un petit serveur créant ainsi une architecture complètement décentralisée où aucun composant central ne peut être attaqué. Cette évolution du réseau BitTorrent vers davantage de décentralisation est similaire à l'évolution réalisée par le client eMule depuis 2004 qui fut conçu initialement pour le réseau eDonkey reposant sur des serveurs avant de supporter le réseau KAD qui est complètement distribué.

Cette décentralisation introduit cependant d'autres problèmes de sécurité bien connus des DHT que nous proposons d'étudier dans ce chapitre. Nous étudions ainsi les vulnérabilités de la principale DHT utilisée par les clients BitTorrent et appelée communément « Mainline DHT » et ce, à travers des expériences à grande échelle menées sur le réseau réel. Nous montrons ainsi que des attaques très efficaces sont possibles et peuvent largement affecter le bon fonctionnement du réseau. Dans un second temps, nous étudions l'applicabilité d'une solution capable de limiter les attaques contre les DHT à celle de BitTorrent.

1. Recording Industry Association of America

Nos contributions visant à améliorer la sécurité du système d'indexation alternatif et distribué de BitTorrent sont les suivantes :

- Nous mettons en évidence des problèmes de sécurité affectant le réseau Mainline DHT
- Nous proposons une architecture distribuée permettant l'évaluation du réseau en conditions réelles.
- Nous analysons et adaptons un ensemble de mécanismes de protection conçus pour KAD [CCF09] afin de combler les failles.

3.2 Architecture de BitTorrent

3.2.1 Architecture historique de BitTorrent

L'architecture de BitTorrent repose sur plusieurs composants que nous décrivons ici :

- Tracker : Entité responsable de l'entre-découverte des pairs en utilisant un serveur central, ou, plus récemment, un service basé sur une DHT.
- Peer : Noeud connecté au réseau, respectivement appelé « Seeder » ou « Leecher » selon qu'il dispose de l'ensemble d'un fichier ou seulement d'une partie.
- Swarm : Groupe de pairs partageant un même fichier. Il est composé de « Seeders » et de « Leechers ».
- Fichier Torrent : Fichier contenant les meta-données décrivant un fichier à diffuser.

Un fichier torrent contient deux principales informations. La première est la liste des serveurs tracker qui sont chargés de mettre en relation les pairs échangeant le contenu associé au torrent, la seconde est la description du contenu, en particulier sa fragmentation en sous-parties ainsi que l'emprunte de chacune d'elle.

Le téléchargement d'un fichier se fait en deux étapes. La première étape consiste en l'obtention du fichier torrent correspondant au contenu désiré. Ce service est proposé par des sites web indexant les fichiers torrent en fonction des mots-clés de leur contenu. La seconde étape consiste a contacter le serveur tracker spécifié dans le fichier torrent de manière à découvrir les autres pairs participant à l'échange du contenu (le swarm) puis à se connecter à eux pour transférer des parties du fichier. En contactant le tracker, le pair courant est automatiquement ajouté à la liste des pairs actifs partageant ce contenu.

L'échange de parties du fichier suit ensuite un processus relativement complexe. Chaque pair dans le swarm peut télécharger depuis n'importe quelle source potentielle. Cependant, afin de sélectionner les pairs à pourvoir, un mécanisme de récompense est utilisé. Ce mécanisme, connu sous le nom de Tit-for-Tat [Coh03] vise à instaurer des transferts équitables au sein du swarm. Ainsi, un pair A va privilégier un pair B si ce dernier lui a déjà transmis des parties du fichier. Une vue globale de l'architecture de BitTorrent est illustrée par la figure 3.1.

3.2.2 La DHT de BitTorrent

Le protocole BitTorrent a acquis de nouvelles fonctionnalités au cours du temps. La plupart sont toujours à l'étude et ne sont pas supportées par l'ensemble des clients. Parmi les extensions du protocole, il y a notamment :

- Distributed Tracker
- Magnet Links

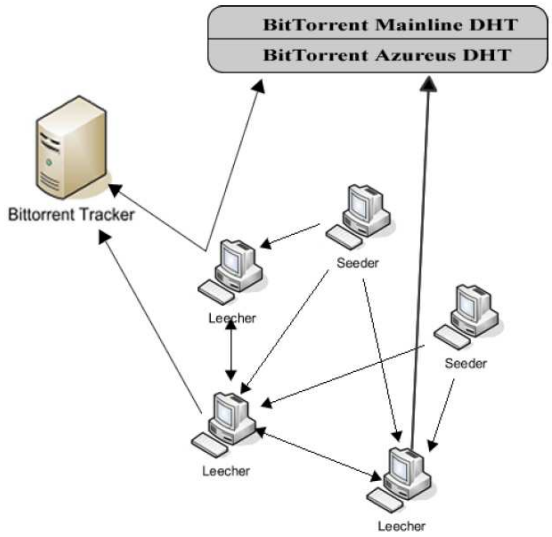

FIGURE 3.1 – L'architecture de BitTorrent

– Multi-Trackers
– Connection Obfuscation

Nous nous intéressons en particulier à l'extension visant à distribuer le service assuré par les serveurs tracker, autrement dit la découverte des pairs constituant un swarm. La liste des pairs peut être obtenue de manière complètement décentralisée en ayant recours aux services d'une DHT au sein de laquelle chaque pair est responsable du référencement de quelques fichiers torrent.

Il y a actuellement deux implantations différentes de cette architecture dans les clients BitTorrent : la DHT d'Azureus qui est propre au client du même nom (désormais renommé Vuze) et la DHT Mainline qui est utilisée par un grand nombre de clients différents parmi les plus populaires (uTorrent, BitTorrent Mainline, BitComet, etc.). Les clients implantant les DHT activent par défaut les services associés. Les deux DHT sont basées sur l'architecture Kademlia [MM02] mais utilisent des protocoles différents les rendant incompatibles. Nous nous limitons à l'étude de la DHT Mainline pour sa plus grande adoption par la communauté.

Le fonctionnement de ces DHT basées sur Kademlia est similaire au fonctionnement de KAD présenté dans le chapitre précédent à savoir que chaque torrent est indexé sur un groupe de pairs dont les identifiants sont proches de celui du torrent qui est obtenu en calculant une empreinte de celui-ci. Les pairs quant à eux choisissent aléatoirement leur identifiant. Le concept de proximité est hérité de Kademlia est définit par la distance XOR entre deux identifiants.

La Figure 3.2 illustre la procédure pour annoncer la participation d'un pair à un torrent et l'obtention des autres pairs participant. Soit le **Pair 9** partageant le fichier **Nirvana**. Il annonce tout d'abord le torrent par un message *Announce* spécifiant qu'il détient ce torrent. **Le pair 52** est responsable de l'indexation des pairs pour ce torrent et doit donc enregistrer le **pair 9** en tant que contact. Le **pair 75** souhaitant télécharger le même fichier envoie un message *GetPeer* au **pair 52** qui répond avec la liste des contacts connus détenant ce torrent et constituant son swarm, incluant le **pair 9**. Le message *GetPeer* va également automatiquement inclure le **pair 75** parmi les contacts de ce torrent. Cette procédure est donc tout à fait similaire à celle

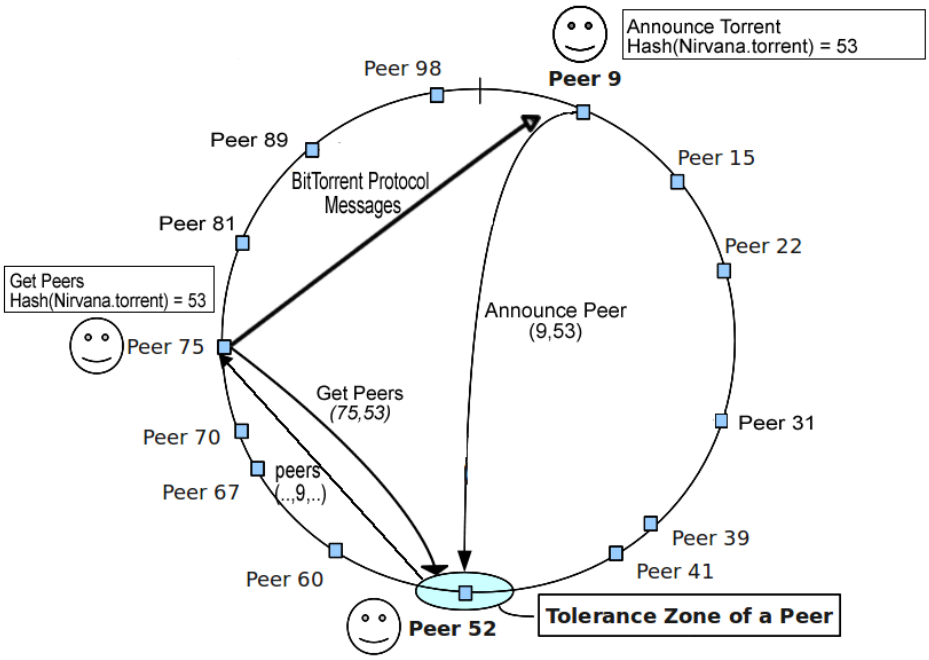

FIGURE 3.2 – Utilisation de la DHT de BitTorrent

reposant sur les serveurs tracker mais l'indexation est distribuée les pairs de la DHT.

Afin de montrer l'utilisation massive des DHT dans le fonctionnement actuel du réseau BitTorrent, nous avons choisi 10 torrents populaires, présentés dans la Figure 3.3 et mesuré dans quelles proportions les contacts obtenus provenaient du serveur tracker ou de la DHT Mainline que nous interrogeons via un plug-in pour le client Vuze. Le client Vuze interroge périodiquement le serveur tracker et la DHT pour obtenir une liste à jour des pairs constituant le swarm. Pendant la durée de l'expérience, 22.834 pairs ont ainsi été trouvés pour l'ensemble des torrents considérés. La Figure 3.4 indique le nombre de pairs découvert pour chaque torrent. La Figure 3.5 montre la proportion des pairs obtenue par chacune des deux méthodes (centralisée ou distribuée) : dans l'ensemble, 70% des pairs découverts ont été obtenus de manière décentralisée. Ce résultat montre clairement que le réseau BitTorrent est désormais parfaitement fonctionnel

Category	Torrent Name	Size	Avg # of Seeders	Avg # of Leechers
Movies	Sex and the City 2 (2010) DVDRip XviD-MAX	1,38 Gb	11708	13837
	Inception.2010.CAM.XviD-TA(NEW SOURCE)	1,42 Gb	8685	6453
Audio	Shakira - Waka Waka - World Cup 2010 Anthem - Time For Africa -	9,45 Mb	5307	496
	Eminem-Recovery-(Retail)-2010-[NoFS]	117,19 Mb	10104	1222
Games	Starcraft.2.Wings.of.Liberty-LiBERTY	7,1 Gb	2425	10151
	The Sims 3 - Razor1911 Final MAXSPEED	5,58 Gb	3114	1928
Applications	ADOBE PHOTOSHOP CS5 EXTENDED EDITION [thethingy]	1,26 Gb	6415	947
	Microsoft Windows 7 Ultimate Retail(Final) x86 and x64	5,34 Gb	2576	2359
Others	Facebook directory - personal details for 100 million users	2,79 Gb	2289	808
	DCP and Minutemen Week of 21-07-10 Complete	5,64 Gb	954	459

FIGURE 3.3 – Torrents utilisés pour mesurer l'utilisation de la DHT

	Torrent
1	Adobe Photoshop CS5 Extended Edition.exe
2	Microsoft Windows 7 Ultimate Retail(Final) x86 (32 bit) and x64 (64 bit)
3	Eminem-Recovery-(Retail)-2010-[NoFS]
4	Shakira_Waka_Waka_World_Cup_2010_Anthem
5	Starcraft.2.Wings.of.Liberty-LiBERTY
6	The Sims 3 - Razor1911 Final MAXSPEED
7	Inception.2010.CAM.XviD-TA
8	Sex and the City 2 (2010) DVDRip XviD-MAXSPEED
9	fbnames
10	DCP.and.Minutemen.Week.of.21-07-10

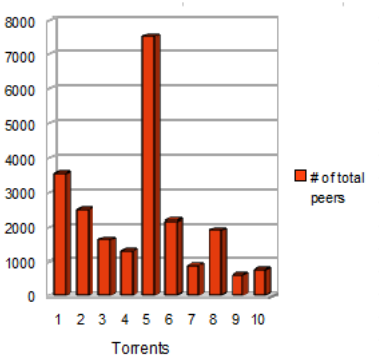

FIGURE 3.4 – Nombre total de pairs vus par fichier torrent

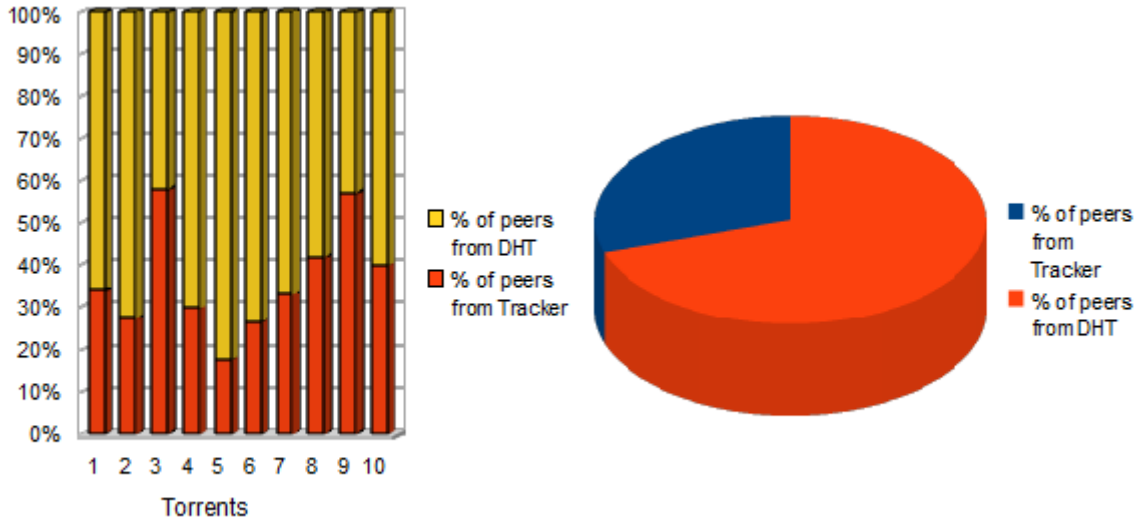

FIGURE 3.5 – Proportion des pairs obtenus de manière centralisée ou distribuée

sans les serveurs tracker et motive d'autant plus notre étude des vulnérabilités de ce nouveau composant distribué puisqu'il tend à devenir l'unique moyen de découverte des pairs.

3.3 Travaux relatifs

BitTorrent a été l'objet de nombreuses études puisqu'il est le protocole pair à pair le plus utilisé ces dernières années. Pour beaucoup d'entre elles, ces études ont visé à superviser le réseau. [LBLF+10] présente ainsi une manière simple mais efficace de superviser l'activité des utilisateurs de BitTorrent souhaitant rester anonymes en exploitant les fuites d'informations vers les DHT. Piatek et al. [PKK08] montrent comment l'exploitation de l'infrastructure de BitTorrent peut permettre facilement de simuler l'implication de n'importe quel équipement réseau connecté à Internet dans un partage de fichier illégal. Saganos et al.[SPR09] analysent

quant à eux un ensemble de torrents populaires afin de détecter et d'éviter les clients déviants. La sécurité de la diffusion de données au sein du réseau a également fait l'objet d'études. Kong et al.[KCW10], [KCWZ10] ont ainsi évalué la pollution de l'indexation sur des serveurs tracker. Ils ont ainsi montré qu'un tracker central peut être pollué de manière à augmenter le temps nécessaire pour rejoindre un groupe de pair connecté appelé « swarm ».

Si l'on considère plus précisément les travaux concernant l'indexation décentralisée, Crosby et al.[CW07] présentent une étude complète sur les implantations différentes de DHT mis en oeuvre par les clients BitTorrent à savoir Mainline et Azureus DHT. Ils étudient ainsi plusieurs aspects tels que la latence et détectent certains problèmes dans l'implantation des algorithmes de routage tout en proposant des améliorations sur la maintenance des tables de routage visant à éviter les noeuds obsolètes. Cependant, leur étude ne s'intéresse pas aux questions de sécurité. Jimenez et al.[JOK09] se sont focalisés sur l'étude des problèmes de connectivité dans ces deux réseaux qui sont principalement dus à l'utilisation de NAT et de pares-feu.

Peu de travaux ont été menés sur la sécurité des DHT utilisées par BitTorrent. Recemment, Jetter et al.[JDH10] ont proposé un mécansime d'auto-enregistrement permettant d'éviter les attaques Sybils sur la DHT de BitTorrent. Ils limitent le nombre de pairs par adresse IP de manière à empêcher l'instanciation de nombreux pairs malveillants depuis une seule machine. Cependant, leur solution est peu pratique car elle brise la rétro-compatibilité entre clients et n'évite pas les attaques distribuées au niveau IP, telles que nous les mettons en oeuvre dans ce chapitre. D'autre part, Wolchok et al. [WH] ont récemment mené une campagne de supervision sur la DHT Azureus. Ils ont montré que le réseau peut être exploré grâce à une attaque Sybil ce qui permet d'étudier les contenus échangés et les comportements des utilisateurs.

Ce chapitre complète le travail de Wolchok et al en analysant cette fois la DHT Mainline et ses problèmes de sécurité.

3.4 Exploitation des vulnérabilités de la DHT

Nous présentons dans cette partie une architecture et un ensemble d'expériences visant à montrer la vulnérabilité de la DHT Mainline face à des attaques internes. Bien qu'implantée depuis 2005, aucune de ses spécifications [Loe08] ne mentionne de mécanisme de protection et aucune étude ne s'est intéressée à ce problème. Les évaluations présentées ci-après reposent sur une architecture distribuée. Même si un seul ordinateur suffit en l'état à réaliser les attaques, quelques règles simples limitant l'influence d'une adresse IP suffirait en effet à limiter des attaques centralisées.

3.4.1 Architecture d'évaluation distribuée

Pour expérimenter des attaques sur la DHT de BitTorrent, nous utilisons l'architecture distribuée illustrée par la Figure 3.6. Celle-ci repose sur nos travaux précédents menés sur le réseau KAD [CCF09].

Nous utilisons un groupe de noeuds de PlanetLab [2] qui exécutent une version modifiée du plug-in utilisé par le client Vuze pour se connecter à la DHT Mainline et qui sont couplés

2. `http://www.planet-lab.org`

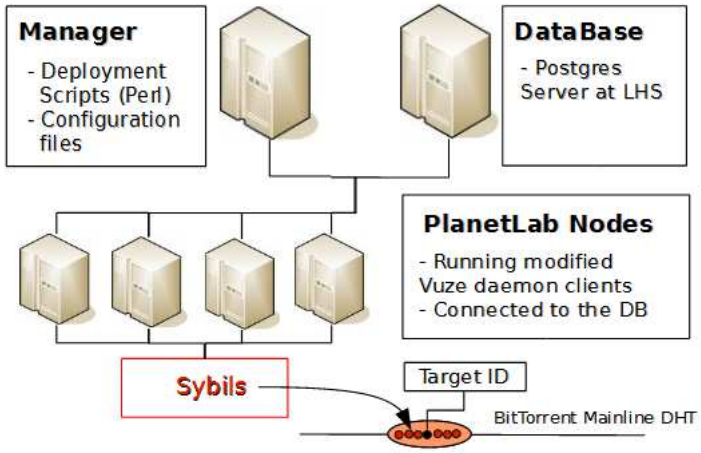

FIGURE 3.6 – Architecture d'attaque distribuée

à une base de données Postgres hébergée au LHS[3]. Cette base contient les paramètres des expérimentations et les données récoltées durant celles-ci. Pour des raisons légales, les clients modifiés ne téléchargent ni ne partagent de fichier, ni même ne rejoignent de swarm[4], leurs communications se limitant à la partie du protocole utilisée pour l'accès à la DHT.

Nous cherchons à exploiter au sein de Mainline une faille bien connue des DHT [UPvS09], à savoir, le libre choix des identifiants des pairs. Cette liberté permet en effet à des pairs malveillants de choisir précisément leur place sur la DHT pour y intercepter les messages. Pour notre expérience, chaque client modifié choisi son identifiant à proximité de l'identifiant d'un torrent, de manière à intercepter les messages dont il est l'objet. Cette attaque est une variation de l'attaque Sybil, déjà expérimentée sur KAD [SENB07a]. Les pairs malveillants sont ainsi appelés « Sybils ».

3.4.2 Expérimentation d'attaques

Nous avons configuré l'architecture de manière à ce que les Sybils partagent entre 110 et 140 bits avec l'ID du contenu ciblé. 110 bits communs sont plus que suffiants pour garantir qu'aucun pair légitime ne soit plus proche de la cible que les Sybils. Par ailleurs, les Sybils collaborent en s'annoncant mutuellement dès que l'un d'entre eux est découvert de manière à attirer l'ensemble des messages visant le torrent ciblé. Etant donnée cette architecture, nous avons réalisé les attaques suivantes :

Supervision du réseau

L'enregistrement de l'ensemble des informations contenues dans les requêtes capturées par les Sybils (adresse IP[5] et ports) permet de connaître les pairs souhaitant obtenir un contenu, et plus globalement, l'activité du contenu supervisé. L'utilisation massive de la DHT du fait de son activation par défaut rend cette méthode de supervision des

3. Laboratoire de Haute Sécurité Informatique, localisé à l'INRIA Nancy
4. Respectant ainsi les conditions d'utilisation de PlanetLab
5. Préalablement anonymisée

contenus et des pairs de plus en plus efficace. Un suivi précis des pairs peut être réalisé grâce à leur identifiant sur la DHT qui est pérenne lorsque les pairs changent leur adresse IP. Cette supervision est peu intrusive est peut être réalisée de manière passive. Bien que l'attaque soit transparente aux utilisateurs, elle permet néanmoins d'obtenir de précieuses informations sur les comportements des utilisateurs ou sur le trafic généré sur la DHT par un torrent particulier.

Nous avons expérimenté cette attaque en déployant 18 Sybils sur la DHT pendant 20 heures autour de l'identifiant d'un torrent de série TV populaire au moment de l'-expérience (Fringe S03E01 [6]). Nous avons ainsi enregistré les requêtes de type *GetPeer* afin de connaître les pairs cherchant à rejoindre le swarm diffusant ce contenu. 1 million de requêtes ont été capturées venant de 91000 adresses IP différentes.

Pollution et éclipse de contenus

Pour cette expérience, nous avons choisi un film populaire (Iron Man 2) et avons déployé la même architecture. Les attaques visant à corrompre l'indexation nécessitent cependant, pour être efficace, la capture de l'ensemble des requêtes visant un torrent par les Sybils. Nous avons dans un premier temps étudié l'influence du nombre de Sybils positionnés autour de la cible sur le nombre de réponses provenant des Sybils reçues par un client normal lors d'une recherche : la Figure 3.7 montre qu'avec 20 Sybils, un client reçoit des réponses des Sybils dans 90% des cas ce qui permet un contrôle du contenu.

La pollution consiste alors à corrompre les réponses avec des adresses IP spécifiques de manière propager l'attaque au sein du Swarm. L'attaque éclipse consiste quant à elle à ne pas répondre lors de la réception d'un message *GetPeer* de telle sorte que le client ne puisse trouver d'autres pairs associés au torrent recherché ce qui empêche l'accès au contenu attaqué. Cette attaque implique que tous les pairs contactés soient des Sybils car si une requête est reçue par un pair légitime, les contacts retournés alors permettront de rejoindre le Swarm.

Durant nos expériences, nous avons été capable de polluer largement un torrent et de l'éclipser de manière intermittente. La raison est qu'en dépit du placement des Sybils, certains pairs moins proches de la cible sont parfois retournés par l'algorithme de routage et permettent la connexion au Swarm.

Eclipse géo-localisée

L'ajout d'une règle supplémentaire lors de la réception de requêtes *GetPeer* permet de réalise une éclipse géo-localisée. Une telle attaque peut être appliquée pour empêcher l'accès à un contenu depuis une certaine zone géographique (par exempple un pays) en fonction des licenses établies. Pour ce faire, une base de données externe donne la correspondance entre une adresse IP et sa localisation géographique et, étant donnée cette dernière, les Sybils décident alors de répondre ou non à la requête et ainsi éclipser le contenu.

Cette discrimination géographique peut également être appliquée à la supervision et l'étude d'un groupe d'utilisateurs particulier.

6. Episode 1 de la saison 3 de la série « Fringe »

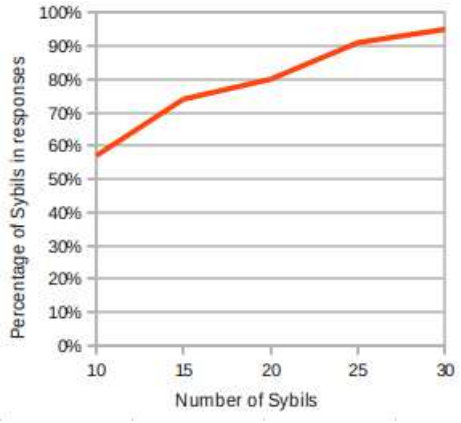

FIGURE 3.7 – Pollution d'un torrent populaire

3.5 Mécanismes de protection

3.5.1 Application des mécanismes de protection de KAD

Comme énoncé précédemment, la DHT Mainline utilisée par BitTorrent repose sur la structure définie par Kademlia. Le réseau P2P KAD, entre autres, est également basé sur Kademlia et fait partie des réseaux P2P les plus déployés aujourd'hui. KAD a cependant inclus un certain nombre de mécanismes visant à rendre le réseau plus résistant à la plupart des attaques connues affectant les DHT. Nous avons évalué ces différents mécanismes dans des précédents travaux [CCF09] et avons montré qu'ils sont efficaces contre les attaques centralisées (menées depuis une seule machine ou depuis un sous réseau), du fait des contraintes imposées sur l'unicité des adresses IP des pairs contactés, mais laissaient le réseau complètement vulnérable à de petites attaques distribuées sur Internet (20 machines). Nous avons ainsi proposé de contrer les attaques localisées en analysant la distribution des pairs sur l'espace d'adressage [CCF10a]. Les avantages des différentes protections expérimentées dans KAD sont d'assurer une parfaite rétro-compatibilité entre les clients protégés et les anciens, contrairement à d'autres propositions [UPvS11] [LSM06] [JDH10], et ce, tout en ayant un surcoût négligeable.

En nous référant à nos précédents travaux sur KAD, nous mesurons la distribution des identifiants au sein de la DHT Mainline afin de savoir si la solution proposée pour KAD est applicable à BitTorrent.

3.5.2 Distribution des identifiants au sein de la DHT Mainline

La détection des attaques par l'analyse de la distribution des identifiants peut être appliquée si, en l'absence d'attaque, les distances entre la cible et les pairs trouvés à l'issue d'un processus de routage suivent une distribution théorique bien déterminée.

La première étape consiste donc à mesurer cette distribution régulière des identifiants sur Mainline. Tout comme pour KAD, les identifiants des pairs légitimes sont choisis aléatoirement. La distance entre un pair et un autre idenfiant est définie par la longueur du préfixe commun (c'est à dire le nombre de bit). Chaque bit en commun supplémentaire avec un identifiant cible divise ainsi par deux le nombre de pairs potentiels correspondant dans le réseau. L'équation

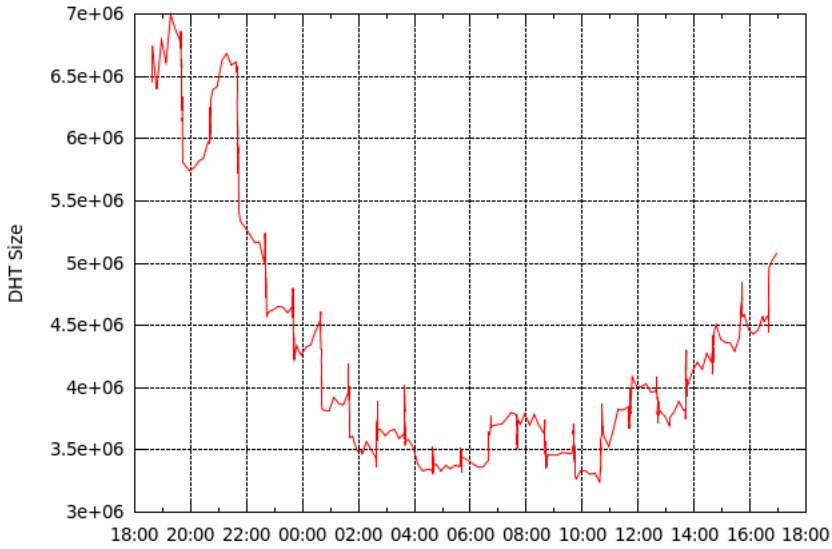

FIGURE 3.8 – Mesure de l'estimation logicielle de la taille de la DHT

(1) décrit le nombre moyen de pairs partageant un préfixe de x bits étant donné un réseau de N pairs.

$$F(x) = \frac{N}{2^x} \tag{3.1}$$

Afin de connaitre le nombre de pairs participant à la DHT Mainline, nous avons relevé périodiquement la taille estimée par un client pendant une journée. La figure 3.8 montre qu'au sein d'une journée la population varie significativement. En omettant les deux premières heures de mesure qui semblent être erronées, nous estimons la population moyenne autour de 4.2 millions de pairs avec cependant une variation du nombre de pairs connectés très importante au sein d'une journée, variant entre 3.2 millions et 6.4 millions.

La distribution théorique étant connue, nous souhaitons savoir si les pairs trouvés à l'issue du processus de localisation de la DHT reflète bien celle-ci. Ceci prouverait que l'algorithme de routage est bien capable de trouver les pairs les plus proches de l'identifiant ciblé.

3.5.3 Mesure des distributions réelles

Pour cela, nous avons réalisé une expérience sur le réseau consistant à enregistrer les 8 meilleurs pairs trouvés par le processus de routage pour des identifiants aléatoires, de sorte à éviter les attaques pouvant affecter les contenus spécifiques. Ainsi, chaque heure, 35 clés aléatoires sont recherchées sur la DHT résultant en 861 recherches. La figure 3.9 montre la longueur moyenne du préfixe des 8 meilleurs contacts trouvés durant l'expérience. Nous pouvons ainsi observer le lien entre la distribution théorique, la distribution mesurée et la taille du réseau. Ainsi, lorsque le nombre de pairs dans le réseau est divisé par deux, la longueur moyenne du préfixe entre deux voisins diminue d'un bit. Nous pouvons en effet observer que la population de la DHT Mainline présentée par la figure 3.8 impacte directement la longueur du préfixe

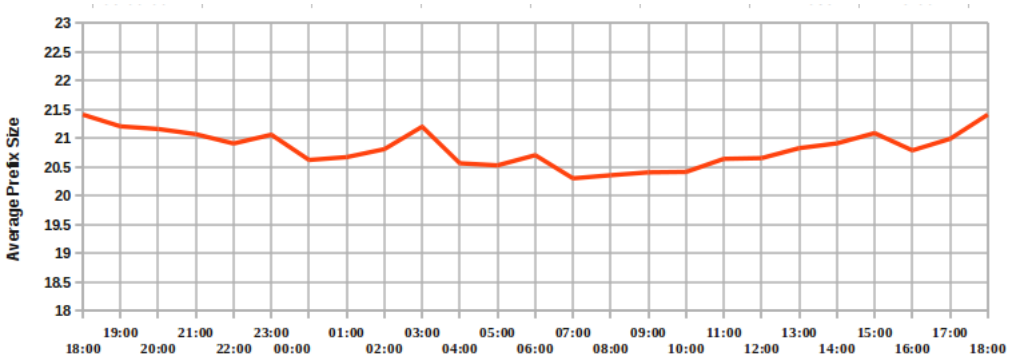

FIGURE 3.9 – Average Prefix Size for the best 8 peers found

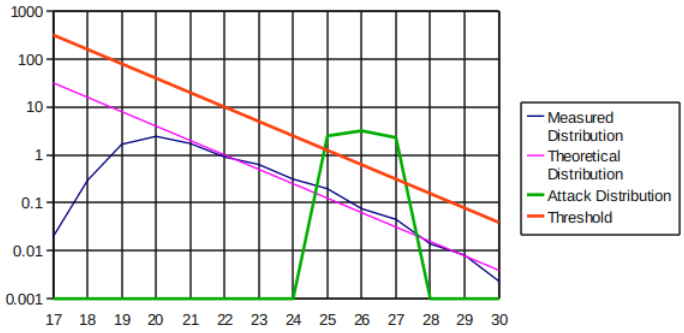

FIGURE 3.10 – Distribution théorique et distribution mesurée

relevé par la figure 3.9. Quand la taille du réseau varie de 3.2 millions de pairs à 6.4 millions de pairs, la taille moyenne du préfixe mesuré varie de 20.5 bits à 21.5 bits.

Après avoir considéré l'ensemble des contacts trouvés et leur distance à la cible, nous calculons le nombre moyen de pairs relevé pour chaque longueur de préfixe. La figure 3.10 présente la distribution théorique pour $N = 4.2$ millions de pairs ainsi que la distribution mesurée des identifiants. Nous pouvons remarquer qu'au delà d'une longueur de préfixe de 20 bits, la distribution mesurée suit parfaitement la distribution théorique ce qui montre que l'algorithme de routage est assez précis pour trouver les plus proches pairs possibles. Afin d'illustrer visuellement le mécanisme de détection, la figure 3.10 montre également un exemple de distribution traduisant une attaque où un attaquant introduit un groupe de Sybils à proximité de la cible (ayant des préfixes de longueur 25, 26 et 27 bits).

3.5.4 Analyse des distributions contre les attaques

Afin d'éviter les pairs qui sont anormalement proches d'une clé recherchée sur la DHT, nous avons proposé une méthode comparant la distribution des idenfiants des pairs trouvés à celle théorique grâce à la divergence de Kullback-Leibler [CCF10a]. Si la distance entre les deux distributions est supérieure à un certain seuil, une attaque est probable. Préalablement à l'application de cette méthode, il est possible de filtrer les pairs les plus suspects en évitant tout

pair partageant plus de 30 bits avec le contenu recherché puisqu'une telle longueur de préfixe est improbable. Le seuil de détection est ensuite obtenu en équilibrant les faux-positifs et les faux-négatifs obtenus en considérant un large spectre d'attaques simulées et des distributions réelles saines.

La distribution des préfixes mesurés suivant la distribution théorique (figure 3.10), la solution proposée pour le réseau KAD est bien applicable à la DHT Mainline. La seule difficulté d'application vient de la plus grande variation de la population au sein d'une journée que présente BitTorrent et qui s'explique par l'utilisation géographique des deux réseaux, en particulier, KAD est équitablement réparti entre l'Europe et la Chine ce qui compense les variations horaires. Nous avons cependant proposé deux méthodes [Cho11] visant à calculer périodiquement la distribution de référence à considérer pour la détection de manière être l'adapater dynamiquement à la population du réseau. La distribution de référence peut ainsi être périodiquement calculée à partir de la population estimée du réseau, ou directement apprise par le processus de routage. Ces optimisations facilitent l'application de notre solution sur la DHT Mainline.

3.6 Conclusion

Nous avons montré dans ce chapitre que la seule distribution des serveurs tracker ne permet pas à BitTorrent de garantir la sécurité et la pérennité des informations échangées. A travers des expériences sur le réseau réel, nous avons en effet montré que des attaques efficaces peuvent être réalisées contre la DHT Mainline en utilisant une architecture distribuée. Actuellement, et en l'absence de mécanismes de sécurité, le système distribué est moins fiable que la solution centralisée initiale puisque quelques noeuds suffisent à polluer ou éclipser un contenu du réseau.

Bien que le principal objectif de ce chapitre est d'amener la comunauté à prendre conscience de ces problèmes, nous sommes également convaincus que ceux-ci peuvent être évités par l'ajout de mécanismes de sécurité au sein de la DHT. Nous avons rappelé à ce propos que plusieurs protections sont possibles et peuvent être rapidement appliquées. La solution la plus efficace consiste à détecter les insertions de pairs malveillants en analysant la distribution des identifiants à proximité d'un contenu par rapport à l'assignation aléatoire régulière que nous avons validée expérimentalement.

Chapitre 4

Détection centralisée des pairs suspects

4.1 Introduction

Nous avons vu dans le chapitre précédent que la principale vulnérabilité pouvant être exploitée pour attaquer les réseaux P2P est l'insertion de noeuds en des positions spécifiques sur la DHT ce qui est nommé par la taxonomie du chapitre 2 en tant qu'attaque interne localisée. Nous proposons dans cette section de détecter les pairs suspects dans le réseau P2P KAD pouvant traduire ce comportement. Pour cela nous réalisons une cartographie du réseau grâce à un explorateur spécifiquement conçu pour obtenir une image très précise de la DHT. Nous analysons ensuite les résultats afin de détecter deux types de positionnements suspects selon qu'ils impliquent localement un groupe de pairs malveillant ou uniquement un seul pair. Nous constatons ainsi pour la première fois la réalité de certaines attaques publiées et pouvons estimer leur nombre au sein du réseau.

Ce chapitre est organisé comme suit : nous présentons tout d'abord les travaux relatifs à l'exploration du réseau KAD. Nous présentons ensuite dans la section 4.2 notre explorateur permettant la découverte des pairs avec une grande précision. Les images ainsi obtenues du réseau sont analysées dans la section 4.3 où deux approches sont utilisées pour détecter les pairs suspects. Enfin, la section 4.4 conclut ce chapitre.

4.1.1 Travaux relatifs à l'exploration des DHT

Un explorateur ou "*crawler*" est un outil capable de découvrir l'ensemble des pairs d'un réseau et de stocker les différentes informations les concernant.

Plusieurs explorations du réseau KAD ont déjà été réalisées à diverses fins. Les auteurs de [WTCT+08] et [SENB07a] découvrent ainsi les pairs du réseau à des fins d'attaque. Pour chaque pair découvert, ils interrogent ce dernier en émettant de nombreuses requêtes de localisation (`Kademlia Request`) vers des identifiants pré-calculés de manière à obtenir tous les contacts de la table de routage du pair interrogé. Ces informations servent ensuite à insérer des Sybils [WTCT+08] ou à corrompre les références de contacts existants [SENB07a]. Utilisant le même explorateur *Blizzard* que [WTCT+08], [SENB07b] réalise des explorations périodiques de la DHT de manière à étudier certaines caractéristiques des pairs dans le temps.

Les auteurs de [YFX+09] utilisent une autre approche basée sur l'interrogation de contacts par des requêtes d'amorçage (bootstrap request). Cette approche est sensée être plus

performante (20 contacts retournés par requête d'amorçage contre 11 pour celle de localisation). Cependant les contacts obtenus sont choisis aléatoirement dans la table alors que les requêtes de localisation spécifient une adresse cible permettant de contrôler le parcours des tables. Les résultats de cette exploration ont mis en évidence un nombre important de pairs (20%) partageant leur identifiant dont les auteurs étudient les causes possibles.

Si de nombreuses observations du réseau KAD ont été réalisées, aucune jusqu'à présent ne s'est intéressée à recenser les attaques pouvant affecter la DHT. De même, aucune étude n'estime l'efficacité de l'explorateur mis en oeuvre dont les algorithmes sont peu détaillés quand ils sont mentionnés.

4.2 Exploration du réseau KAD

4.2.1 Méthode d'exploration

La conception de notre explorateur vise deux objectifs. D'une part, obtenir une vision précise du réseau, et d'autre part, limiter l'empreinte de l'exploration sur le réseau en limitant le nombre de requêtes envoyées à chaque pair. Ceci permet en outre d'obtenir une exploration compatible avec les limitations implantées dans les derniers clients, contrairement aux précédentes stratégies d'exploration désormais limitées, notamment par rapport à la protection contre l'inondation empêchant un pair de recevoir rapidement des messages d'une même source.

Notre méthode d'exploration se divise en trois phases décrites ci-après.

Amorçage

La phase d'amorçage sert à obtenir une première image imprécise de l'ensemble de la DHT. Pour cela, des requêtes d'amorçage (`Bootstrap`) sont émises. Les requêtes d'amorçage permettent d'obtenir 20 contacts tirés aléatoirement dans la table de routage du pair sollicité et sont donc parfaitement adaptées à une première découverte globale de la DHT. De nouveaux contacts sont ainsi progressivement interrogés au fur et à mesure des réponses jusqu'à ce que 500000 contacts aient été découverts dont au moins 500 par zone [1]. Au delà de cette valeur, les contacts retournés étant sélectionnés au hasard, il est de plus en plus difficile d'apprendre de nouveaux contacts par cette méthode.

Exploration complète

Ensuite, chaque zone est explorée avec précision grâce aux requêtes de localisation (`Kademlia Request`). Un pair ainsi interrogé retourne les 4 contacts les plus proches connus de l'identifiant spécifié en paramètre. Afin de découvrir l'ensemble des pairs, nous générons $2^{21} \approx 2$ millions de « KADIDs cibles » uniformément répartis et envoyons pour chacun d'eux une requête de localisation au pair le plus proche déjà découvert. Ainsi, $2^{13}(2^{21}/2^8)$ KADIDs cibles sont générés dans chaque zone selon le format :

1. une zone est une subdivision artificielle de l'espace d'adressage basée sur le premier octet de poids fort des identifiants (de *0x00* à *0xFF*)

$$\underbrace{ZZZZZZZZ}_{8\ bits\ de\ la\ zone}\ \overbrace{FFFFFFFFFFFFF}^{13\ bits\ fixes\ de\ 0\ a\ 2^{13}-1}\ \underbrace{RRRRRR...R}_{107\ bits\ aleatoires}$$

où Z, F et R désignent respectivement des bits de zone, les bits fixés et ceux tirés aléatoirement une fois.

Seconde passe

Dès qu'une zone a été explorée, c'est à dire quand tous les KADIDs cibles de cette zone ont été envoyés, une seconde exploration de celle-ci a lieu pour en améliorer la cartographie. Pour chaque contact précédemment découvert, on calcule alors son voisin le plus proche dont on extrait ensuite le préfixe commun de longueur x bits entre les deux KADIDs. On construit ensuite un nouveau « KADID cible » partageant ce préfixe et où les $(128-x)$ bits restants sont aléatoires. Une requête de localisation pour ce KADID cible est finalement envoyée au contact. Cette phase permet de découvrir quelques contacts manqués lors de l'exploration complète. L'exploration se termine lorsque tous les contacts ont ainsi été interrogés sur leur voisinage immédiat.

4.2.2 Cartographie obtenue

Informations enregistrées

Pour chaque pair découvert, nous enregistrons les informations suivantes ¡KADID, adresse IP [2], port TCP, port UDP, version de KAD, état du pair¿. La version de KAD fait référence à la version du protocole implantée par le client, l'état du pair est P(*possible*), T(*tried*) ou R(*responded*) selon respectivement que le contact a juste été découvert, a été contacté ou a répondu.

```
[...]
<32FFF76959F6A7095347FB338B304330, #.#.#.#, 38060, 16905, 0, T>
<32FFFC5C4D5AE9A082871FF68B1F0D9C, #.#.#.#, 5149, 1025, 4, R>
<32FFFC5C4D5AE9A082871FF68B1F0D9C, #.#.#.#, 5149, 5159, 4, P>
Zone 33: 15196 contacts
<3300048A90460A8AAC3DD2FF542ADF98, #.#.#.#, 12399, 39949, 9, R>
<3300083A0480CFA91B8C142401DD26F2, #.#.#.#, 5611, 5621, 8, T>
<330018506569424D7CBA7133F437EDC8, #.#.#.#, 6647, 6657, 8, P>
<33002596F7AAAA4348FB4349F0A14FA4, #.#.#.#, 46318, 61632, 9, R>
<33002EF905E27753B1900BC602D29C20, #.#.#.#, 19774, 19774, 8, T>
<33004546934FABE9685674DE1598548F, #.#.#.#, 51478, 52073, 9, R>
[...]
```

Résultats généraux

L'exploration d'une zone compte entre 13000 et 17000 contacts, le nombre total de pairs mesuré allant de 3,3M à 4,3M selon le jour et l'heure de l'exploration. D'un point de vue

2. certaines adresses IP sont anonymisées dans le cadre de ce rapport

macroscopique, la répartition des pairs sur l'ensemble de l'espace d'adressage de la DHT est bien uniforme (figure 4.1), conformément à ce qu'on peut attendre de la majorité des pairs légitimes générant aléatoirement leur identifiant à la première connexion.

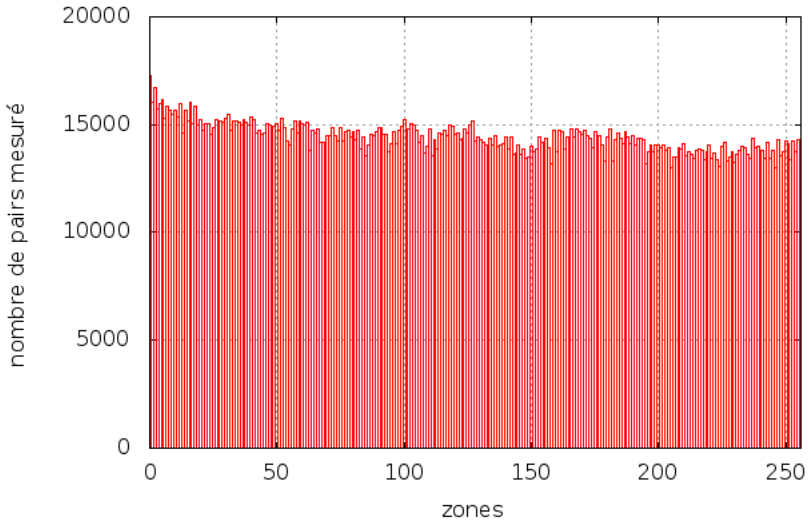

FIGURE 4.1 – Répartition des pairs sur la DHT

Nous analysons plus précisément les résultats d'exploration dans la section suivante, avec pour objectif de détecter les placements traduisant des comportements déviants. Les résultats obtenus pour les différentes explorations réalisées étant similaires, la suite de ce chapitre utilise les données d'une exploration réalisée le 8 Juillet 2010 et comptant 3688932 pairs.

4.2.3 Évaluation

Nous avons évalué notre explorateur de deux façons. Nous avons tout d'abord injecté 360 pairs dans KAD suivant une configuration d'attaque depuis l'infrastructure d'expérimentation distribuée PlanetLab [3]. Les Sybils sont ainsi répartis par groupe de 5 sur 72 identifiants cibles dont ils partagent au moins 96 bits. A l'issue d'une exploration du réseau concomitante à l'attaque, la totalité des pairs insérés était bien présente dans les résultats de l'exploration. L'extrait ci-dessous montre une analyse des données recherchant les pairs à proximité d'identifiants donnés en paramètre (ici les 72 identifiants ciblés).

```
[...]
KADID 71: 19856E29730F11CA0E0C210630ADCB36
<19856E29730F11CA0E0C210621142E70, 62.108.171.74, 14337, 13602, 8, T> [prefix = 99]
<19856E29730F11CA0E0C2106546F8C89, 193.167.187.186, 14690, 13799, 8, T> [prefix = 97]
<19856E29730F11CA0E0C21065622F60F, 155.245.47.241, 13953, 13779, 8, T> [prefix = 97]
<19856E29730F11CA0E0C210676E74885, 212.51.218.235, 13897, 14465, 8, T> [prefix = 97]
```

3. http ://www.planet-lab.org/

```
<19856E29730F11CA0E0C21069636476A, 129.97.74.14, 14308, 13853, 8, T> [prefix = 96]
KADID 72: EBCBA6D72037ED01F56809A9FFE6A86E
<EBCBA6D72037ED01F56809A9268DA7FB, 155.245.47.241, 13915, 13842, 8, T> [prefix = 96]
<EBCBA6D72037ED01F56809A94519B1D4, 129.97.74.14, 14029, 13914, 8, T> [prefix = 96]
<EBCBA6D72037ED01F56809A9702F72B7, 193.167.187.186, 13666, 14427, 8, T> [prefix = 96]
<EBCBA6D72037ED01F56809A9892C91A4, 62.108.171.74, 13853, 14683, 8, T> [prefix = 97]
<EBCBA6D72037ED01F56809A9BAD2A19E, 212.51.218.235, 13861, 13939, 8, R> [prefix = 97]

72/72 of the proposed KADIDs are targeted with at least 96 bits by:
37 IP addresses (showing 361 unique KADIDs in the whole crawler's data)
21 subnets /24  (showing 362 unique KADIDs in the whole crawler's data)
```

Une seconde évaluation a consisté à modifier un client KAD afin d'afficher la liste des contacts trouvés lors d'une publication et à explorer conjointement la zone correspondante. L'ensemble des pairs trouvés par le client aMule l'a également été par l'explorateur, ce qui tend également à montrer l'efficacité de notre exploration.

4.3 Détection des pairs suspects

Comme expliqué précédemment, une attaque sur la DHT implique l'insertion d'un ou plusieurs pairs à proximité de l'identifiant ciblé, afin d'attirer tout ou partie des requêtes à son attention. Pour une meilleure efficacité, plusieurs pairs peuvent être insérés conjointement afin d'attirer davantage de requêtes.

4.3.1 Détection par densité des pairs

Notre première analyse s'intéresse à localiser de tels groupes de pairs sur la DHT. Nous cherchons ainsi à détecter les couples de pairs dont la distance trop proche traduit un placement intentionnel à proximité d'un tierce identifiant plutôt qu'un choix aléatoire de leurs identifiants.

$$F(x) = \frac{N}{2^x} \tag{4.1}$$

Soit F la fonction donnant le nombre moyen de pairs partageant x bits avec un pair courant étant donné un nombre total N de pairs dans le réseau. Nous considérons un nombre de 4 millions de pairs connectés simultanément. Le tableau 4.1 en présente certaines valeurs pour $N = 4 \times 10^6$ et $x \in [1; 128]$. De plus le préfixe moyen partagé entre deux pairs consécutifs est de $d_{moy} = log2(N) = 21.93$ bits.

Étant donné notre exploration de la DHT, nous avons calculé le préfixe commun entre chaque pair et son plus proche voisin, les résultats sont présentés par la figure 4.2. Si les préfixes jusqu'à 35 bits sont communément partagés entre voisins et ne permettent pas de détecter les attaques, les contacts partageant davantage de bits traduisent un placement intentionnel. Le premier graphe de la figure 4.3 illustre cette déviation de la norme théorique (équation 1) pour les contacts partageant entre 22 et 45 bits. Plus que le préfixe commun est élevé, plus l'espérance de trouver de tels voisins est faible et traduit un placement intentionnel ce qui est illustré par le second graphe de la figure 4.3. Nous avons ainsi relevé 426 groupes de contacts anormalement

Nombre de bits en commun	Nombre moyen de pairs
1	2,000,000
8	15625
12	976.5
16	61
18	15,25
20	3.8
24	0.24
28	0.015
32	$9.32 * 10^{-4}$
64	$2.17 * 10^{-13}$
96	$5.05 * 10^{-23}$
128	$1.17 * 10^{-32}$

TABLE 4.1 – Nombre moyen de pairs partageant un préfixe avec un identifiant donné pour une DHT de 4 millions

proches (partageant un préfixe entre 35 et 127 bits) et traduisant autant d'attaques groupées potentielles.

Nous avons par ailleurs réalisé cette analyse sur une seconde exploration de KAD effectuée en avril 2011 et pour laquelle 2074 groupes de pairs suspects ont pu être mis en évidence. Les groupes d'attaquants montrent en outre des motifs d'attaque évidents en partageant un préfixe identique (40 bits), en utilisant des adresses IP appartenant au même sous réseau ou encore des ports spécifiques. L'exemple ci-dessous illustre deux groupes de ces pairs. Ceci tend à prouver que les attaques affectant le réseau évoluent dans le temps, les résultats d'explorations éloignées dans le temps étant différents.

```
Prefix "4A9D8C877700000000000000000000000", length 40, shared by 6 contacts:
<4A9D8C87774AF8C551FE78BDDC3F5A37, 123.144.174.128, 10875, 10875, 8, T>
<4A9D8C877780DFB9985E75EE92AD1C68, 123.144.160.21, 10875, 10875, 8, T>
<4A9D8C877780DFB9985E75EE92AD1C68, 123.145.184.122, 10875, 10875, 8, T>
<4A9D8C877797D58D4C21B5BD5224F067, 123.144.160.98, 10875, 10875, 8, T>
<4A9D8C877797D58D4C21B5BD5224F067, 123.144.167.199, 10875, 10875, 8, T>
<4A9D8C8777F0F03BD1FE123548E269D2, 123.144.163.209, 10839, 10839, 0, R>

Prefix "4A9D8C8777800000000000000000000000", length 41, shared by 4 contacts:
<4A9D8C877780DFB9985E75EE92AD1C68, 123.145.184.122, 10875, 10875, 8, T>
<4A9D8C877797D58D4C21B5BD5224F067, 123.144.160.98, 10875, 10875, 8, T>
<4A9D8C877797D58D4C21B5BD5224F067, 123.144.167.199, 10875, 10875, 8, T>
<4A9D8C8777F0F03BD1FE123548E269D2, 123.144.163.209, 10839, 10839, 0, R>
```

Cependant, quelque soit l'exploration, l'écart le plus important concerne le préfixe de 128 bits (1 million de pairs) qui correspond aux pairs partageant exactement le même identifiant et mérite une analyse à part.

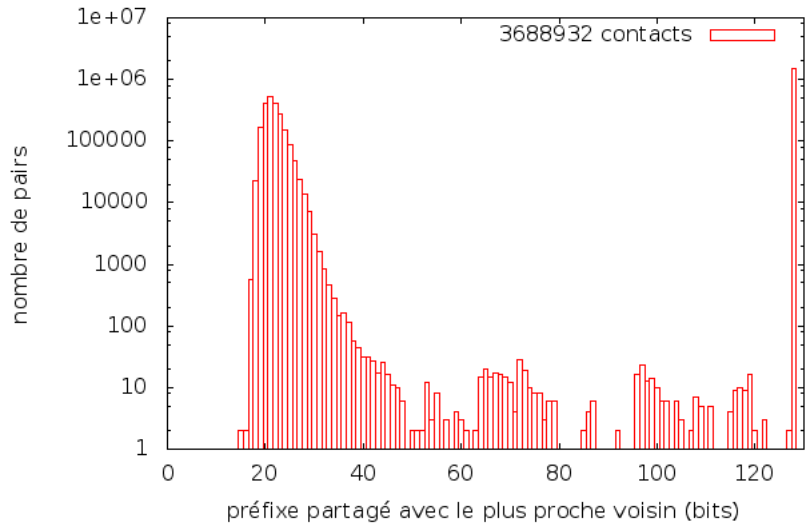

FIGURE 4.2 – Répartition des préfixes entre voisins sur la DHT

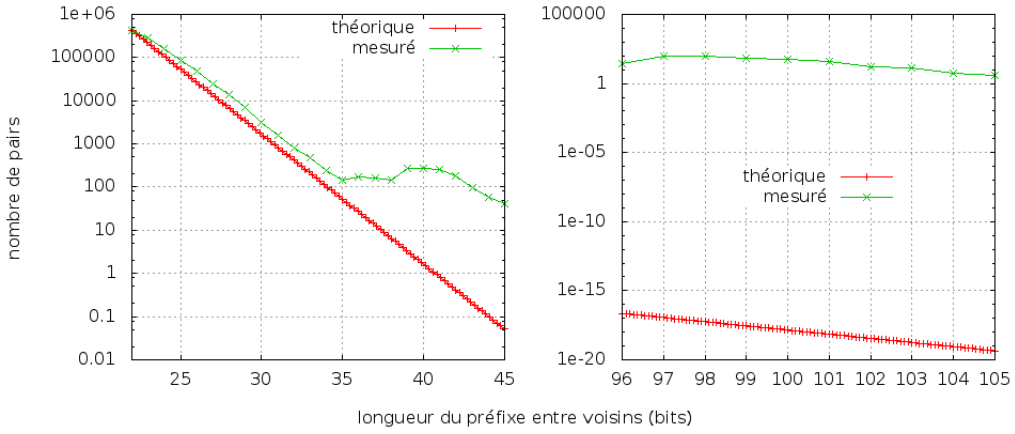

FIGURE 4.3 – Nombre moyen théorique et mesuré de pairs en fonction du préfixe partagé

Identifiants partagés

En effet, sur les 3688932 pairs trouvés lors de l'exploration, on ne dénombre après analyse que 2613963 KADIDs différents. Tout comme [YFX+09], nous constatons donc l'existence de KADIDs partagés par plusieurs pairs. Plus précisément, parmi les KADIDs relevés :

- 82,36% (2152900) des KADIDs sont utilisés par un pair unique,
- 17.64% (461063) des KADIDs sont partagés par plusieurs pairs dont :

– 10,42% des KADIDs sont commun à 2 pairs,
– 2,85% des KADIDs sont commun à 3 pairs,
– les pourcentages décroissant jusqu'à 1 KADID partagé par 259 pairs.

Le partage de préfixe peut traduire une attaque si plusieurs pairs sont insérés exactement avec l'identifiant de la cible. Cependant, il peut également traduire un changement bénin de configuration d'un pair. En effet, un pair changeant d'adresse IP (allocation dynamique d'adresse, mobilité), ou de port de communication durant sa connexion au réseau apparaîtra deux fois avec le même identifiant le temps que la DHT mette à jour ses références. Afin d'éviter de compter ces cas, nous supprimons de la liste des identifiants suspects les cas pour lesquels deux pairs partagent un identifiant tels que seule l'adresse IP ou seul le port changent entre les deux pairs. Ainsi parmi les KADIDs partagés entre deux pairs (272149) :

– 49.73% ne diffèrent que par l'adresse IP (ports UDP et TCP identiques),
– 26.91% ne diffèrent que par le port UDP,
– 1.44% ne diffèrent que par le port TCP,
– 21.92% sont suspects.

Par cette méthode, 248569 identifiants différents peuvent être suspectés. Malgré les précautions prises, ce chiffre peut être soumis à des faux positifs. Nous proposons une dernière estimation plus fiable des attaques affectant KAD, car basée sur les contenus et non uniquement sur les pairs.

4.3.2 Détection par proximité aux ressources

Les analyses précédentes ont une limite importante : elles permettent d'identifier des attributions d'identifiants suspects sans pour autant pouvoir les corréler à un contenu précis. Par ailleurs, les analyses précédentes étant basées sur des proximités entre pairs, au moins deux pairs doivent être insérés pour être détectés, les attaques impliquant qu'un pair passant inaperçues.

Une manière fiable de détecter les attaques est donc de pouvoir mettre en évidence la proximité anormale des pairs malveillants par rapport à une ressource plutôt que la proximité des pairs entre eux. La difficulté de cette approche est que les identifiants des ressources ne sont pas connus à priori. Pour appliquer cette méthode, nous avons extrait des mots-clés de contenus pouvant être partagés sur KAD depuis plusieurs sources d'information (meilleures ventes Amazon, iTunes, fichiers populaires sur ThePirateBay). Nous avons ensuite calculé l'identifiant de chacun des mots-clés composant les différents titres par la fonction MD4 utilisée par KAD. Nous avons finalement recherché les contacts étant anormalement proches de ces identifiants (partageant un préfixe supérieur à 30 bits) dans les données des explorations. Un extrait des résultats est donné ci-après.

```
[...]
twilight 4D62D26BB2A686195DA7078D3720F60A
<4D62D26BB2A686195DA7078D3720F632, X.Y.#.#, 7290, 7294, 8, R> [prefix = 122]
soundtrack AC213377BB53F608390BD94A6AE6DD35
<AC213377BB53F608390BD94A82582F42, #.#.#.#, 5003, 5002, 8, R> [prefix = 96]
harry 770CF5279AB34348C8FECF9672747B94
<770CF5279AB34348C8FECF96524D8CDE, #.#.#.#, 5003, 5002, 8, P> [prefix = 98]
robin B9DF47E5BFAD75F8EE5E3F50EA217983
<B9DF47E5BFAD75F8EE5E3F5051F34AA8, #.#.#.#, 5003, 5002, 8, R> [prefix = 96]
```

```
<B9DF47E5BFAD75F8EE5E3F50EA21799F, X.Y.#.#, 7290, 7294, 8, R> [prefix = 123]
[...]

216/888 of the proposed keywords are targeted with at least 96 bits by:
44 IP addresses (showing 2119 unique KADIDs in the whole crawler's data)
41 subnets /24  (showing 2155 unique KADIDs in the whole crawler's data)
```

Sur les 888 mots-clés utilisés pour cette analyse, un quart d'entre eux avaient un pair proche partageant au moins 96 bits ce qui, étant donné l'espérance de trouver un pair légitime avec un tel préfixe (voir tableau 4.1) traduit sans équivoque un placement intentionnel et un comportement malveillant. Un échantillon de ces mots-clés est donné dans le tableau 4.2, certains faisant référence à un contenu explicite, d'autres étant plus génériques.

mot-clé	meilleur préfixe
avatar	126
invictus	123
sherlock	122
princess	122
frog	98
ncis	96
nero	96
...	...

mot-clé	meilleur préfixe
nine	122
love	122
american	97
russian	97
the	96
black	96
pirate	96
...	...

TABLE 4.2 – Exemples de mots-clés attaqués

Pour les pairs malveillants ainsi détectés, nous avons recherché leur présence sur l'ensemble de la DHT afin de découvrir d'autres identifiants ciblés et absents de la liste initiale de mots-clés. Nous avons ainsi relevé que les seuls mots-clés recherchés ne représentent que 10% de la présence de ces clients (adresse IP + port) sur la DHT. En comptant les 216 identifiants de mots-clés initiaux, ces clients sont au total présents sur 2119 KADIDs. Ce résultat montre clairement que de nombreux contenus de la DHT sont attaqués, parmi les plus populaires. De plus, des configurations d'attaques émergent rapidement des données. Par exemple, parmi les 216 identifiants, 205 sont ciblés par des pairs ayant exactement les ports suivants : UDP=5003, TCP=5002, un préfixe de 96bits mais des adresses IP distribuées sur plusieurs réseaux. Un autre attaquant cible 16 identifiants parmi les 216 en utilisant des pairs ayant exactement les ports : UDP=7290, TCP=7294, un préfixe de 122bits et une adresse IP venant d'un sous réseau spécifique (16 de la forme `X.Y.#.#`).

Bien que cette estimation soit fiable, elle a également des limites, notamment quant au jeu de caractères utilisé par les mots-clés. Ceux considérés pour notre expérience utilisent en effet l'alphabet latin, or, KAD est pour moitié utilisé en Asie. Les pairs ciblant spécifiquement des contenus décrits avec des caractères asiatiques peuvent échapper à cette analyse. Par ailleurs d'autres attaques peuvent cibler exclusivement les fichiers et non les mots-clés.

4.4 Conclusion

Alors que plusieurs attaques pouvant affecter le réseau KAD ont été décrites dans de précédents travaux et que de nombreuses observations de ce réseau ont déjà été réalisées, aucune d'entre elles ne s'était intéressée jusqu'alors aux questions de sécurité affectant la DHT. Afin d'estimer les positionnements anormaux des pairs pouvant traduire des attaques, nous avons tout d'abord développé et évalué un explorateur capable de découvrir précisément la DHT de KAD, malgré les limitations récemment incluses dans les clients.

Une première analyse considérant la proximité entre les identifiants des pairs a mis en évidence des regroupements de pairs anormaux, quelques pairs étant trop proches les uns des autres (426 en juillet 2010, 2074 en avril 2011) mais la grande majorité d'entre eux partageant un même identifiant (248569). Une seconde analyse basée sur l'étude de mots-clés populaires a mis en évidence qu'une grande proportion de ceux-ci est attaquée. Les pairs impliqués sont d'ailleurs présents sur de nombreux identifiants de la DHT (2119) et des configurations d'attaques peuvent être clairement mises en évidence. Concernant les mots-clés ciblés, les attaquants insèrent un seul pair extrêmement proche du contenu (96 bits ou 122 bits communs) mais ne semblent en revanche pas réaliser d'attaques impliquant plusieurs pairs.

Les deux approches de détection sont de plus complémentaires. La première, basée sur l'analyse des distances inter-pairs, permet une détection des attaques sans nécessiter la connaissance des contenus ciblés mais ne détecte que des attaques massives (i.e. où plusieurs pairs sont insérés). La seconde, basée sur l'analyse des distances pairs-contenus, permet de détecter des attaquants isolés mais nécessite la connaissance a priori du contenu ciblé.

Dans le cadre du projet Acda-P2P, la suite de ces travaux consiste (1) à détecter d'autres formes de comportements suspects, tels que présentés dans le chapitre 1, et (2) à étudier plus précisément les pairs suspects ainsi mis en évidence. Nous devons pour cela observer dans un premier temps d'autres paramètres du réseau P2P. Une fois l'ensemble des comportements suspects supervisés, communiquer avec les pairs détectés via les primitives du protocole KAD permettra de mieux identifier leur comportement (surveillance, déni de service, pollution...) et leurs moyens de mise en œuvre. Cette connaissance doit permettre le développement de sondes autonomes capables de détecter les pairs déviants à l'issue du projet Acda-P2P.

Le chapitre suivant présente les résultats d'une campagne de mesure allant dans ce sens en suivant à l'évolution des positionnement anormaux à long terme et en corrélant ceux-ci à la popularité des contenus.

Chapitre 5

Caractérisation des attaques pour les contenus populaires

5.1 Introduction

Le présent chapitre décrit une collecte de données visant à mieux caractériser les pairs ciblant des contenus populaires. Nous présentons dans la section 5.2 l'architecture de supervision mise en oeuvre qui inclut deux composants : d'une part la découverte des contenus populaires par consultation d'une base de données multimédia, et d'autre part l'observation des pairs ciblant ces contenus au sein de la DHT du réseau KAD. Nous analysons ensuite, dans la section 5.3, les résultats obtenus après un mois de collecte et les comparons les mesures obtenues à celles présentées dans le précédent chapitre. Nous constatons l'absence d'attaques durant cette seconde période de mesures ce qui empêche l'exploitation souhaitée des résultats.

5.2 Architecture de mesure

5.2.1 Consultation d'une base de donnée multimédia

La première étape consiste à interroger quotidiennement la base de donnée du site commercial Amazon via son service web « Product Advertising API [1] ». Celui-ci permet de lister, avec certaines restrictions, les produits proposés par le site de vente en ligne et leurs propriétés, notamment leur classement en terme de ventes « SalesRank » renseignant sur la popularité du produit à un moment donné. Nous explorons ainsi les produits appartenant à deux types de contenus, à savoir les types « DVD » et « Music », grâce à un programme Java qui interroge le web service et enregistre les informations concernant les produits trouvés dans notre base de données. Parmi l'ensemble des produits, nous enregistrons en particulier ceux appartenant au top 100 de chaque catégorie (c'est à dire très populaires) et d'autres à intervalles suivant une échelle logarithmique. Notre objectif est ainsi de pouvoir appréhender la probabilité qu'un contenu soit attaqué en fonction de sa popularité. Finalement, l'ensemble des contenus enregistrés dans la base de données est mis à jour quotidiennement puis utilisé pour orienter la supervision

1. http ://docs.amazonwebservices.com/AWSECommerceService/2010-11-01/DG

des attaques sur le réseau KAD. La popularité des différents contenus est suivie dans le temps puisque les contenus ajoutés ne sont jamais supprimés de notre base.

La majeur difficulté de cette étape réside dans le fait que le service web d'Amazon ne permet pas d'obtenir les contenus strictement ordonnés selon leur popularité. Pour chaque interrogation, un sous-ensemble aléatoire des produits est retourné. La multiplication des requêtes permet de reconstruire le top 100 et d'obtenir des produits suivant l'échelle logarithmique, sans toutefois garantir la constance des rang obtenus d'une exécution à l'autre et au prix d'une surcharge importante du web service. A l'issue de la consultation des produits, un fichier XML est généré contenant pour chaque produit son nom et son identifiant dans la base. D'autres informations sur les produits sont enregistrés dans la base de donnée comme son producteur.

```
<Item>
<id>47951</id>
<Title>
The Lord of the Rings: The Motion Picture Trilogy (Extended Edition + Digital Copy)
[Blu-ray]
</Title>
</Item>
<Item>
<id>47952</id>
<Title>
Harry Potter and the Deathly Hallows, Part 1
</Title>
</Item>
```

5.2.2 Mesures sur le réseau P2P KAD

L'ensemble des mots-clés extraits des noms des produits obtenus d'Amazon est ensuite utilisé pour guider la supervision quotidienne du réseau KAD. L'empreinte MD4 de chaque mot-clé est ainsi calculée et permet d'obtenir l'identifiant du mot-clé dans le réseau P2P. Un client KAD modifié collecte les informations sur les 10 pairs les plus proches des mots-clés à étudier en sollicitant le processus de localisation pour chacun des identifiants. Chaque jour, un mot-clé est supervisé autant de fois qu'il apparaît dans les noms des produits à superviser, de nombreux mots-clés populaires tels que « avatar », « harry », etc. présents font donc l'object de plusieurs mesures car ils sont associés à plusieurs produits multimedia. A l'issue de la supervision quotidienne, un fichier XML est généré contenant pour chaque mots-clés les pairs trouvés. Le schéma 5.1 représente l'ensemble de l'architecture de collecte des données.

```
<Keyword>
<Node>
<Common_prefix>24</Common_prefix>
<IP>123.119.157.222</IP>
<KADID>59DE740794A317AC37167219E2877BFB</KADID>
</Node>
<Node>
<Common_prefix>20</Common_prefix>
<IP>122.235.181.124</IP>
```

```
<KADID>59DE7DCC6F8D2CF51E438AF4D12FCD45</KADID>
</Node>
<Node>
<Common_prefix>20</Common_prefix>
<IP>83.31.19.18</IP>
<KADID>59DE7E6B419A6A0CF98E28254B15B2BC</KADID>
</Node>
<Node>
<Common_prefix>19</Common_prefix>
<IP>93.145.30.235</IP>
<KADID>59DE60554F52EA737EFFF6BB2D326D07</KADID>
</Node>
[...]
<Node>
<Common_prefix>17</Common_prefix>
<IP>222.95.47.147</IP>
<KADID>59DE234AA41B55388F782553DA0473CE</KADID>
</Node>
<Name>inception</Name>
<KWDID>59DE74EB8935501D6F92C9BEFB603040</KWDID>
<Attack>no</Attack>
</Keyword>
```

5.3 Analyse des données

5.3.1 Données collectées

Nous analysons dans cette section les données récoltées durant une période d'un mois allant du 21 février 2011 au 21 mars 2011. Notre procédure de collecte gardant l'historique des contenus précédemment supervisés afin d'apprécier l'évolution de leur popularité (et donc potentiellement des attaques associées) dans le temps, le nombre de contenus à superviser chaque jour est strictement croissant. Ceci entraîne une augmentation de la taille des données collectées. Ainsi au 21 février, 636 références furent observées générant 2207 mots-clés et 16.7 Mo de données de supervision alors qu'au 21 mars, 2433 références furent observées générant 9062 mots-clés et 180.1 Mo de données.

L'analyse des données montre cependant que très peu de pairs suspects sont placés autour des mots-clés obtenus d'Amazon. Conformément à nos précédents travaux [CCF10a], nous considérons dans un premiers temps que les pairs partageant plus de 28 bits avec l'identifiant d'un mot-clé sont peu probables et par conséquent suspectés de s'être placés sciemment dans la DHT. Les graphiques 5.2 et 5.3 illustrent le très faible nombre de mots-clés potentiellement attaqués au regard du nombre de mots-clés supervisés. Le graphique 5.3 montre cependant quelques mots clés potentiellement attaqués et qui doivent être étudiés plus précisément.

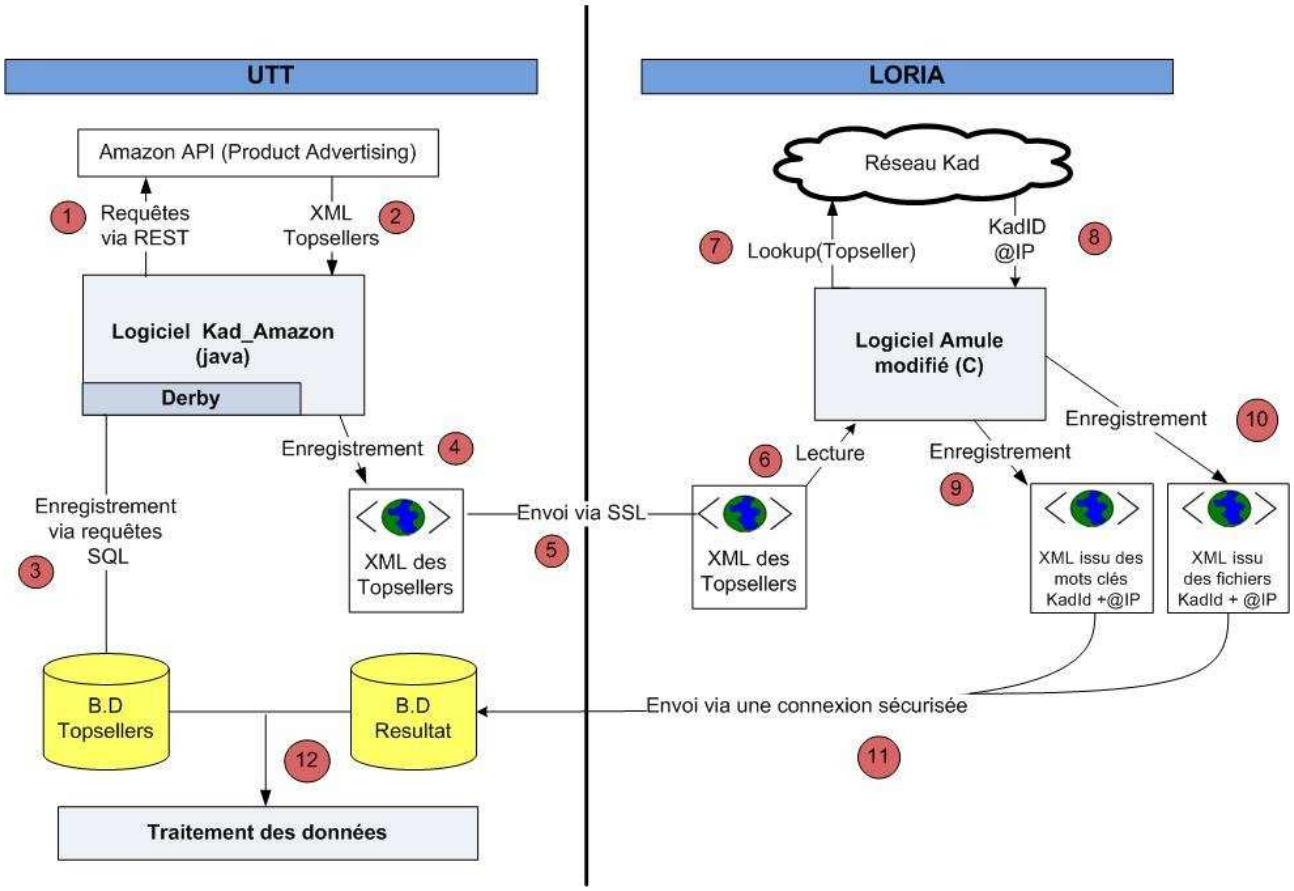

FIGURE 5.1 – Mise en œuvre générale du projet KAD-Amazon

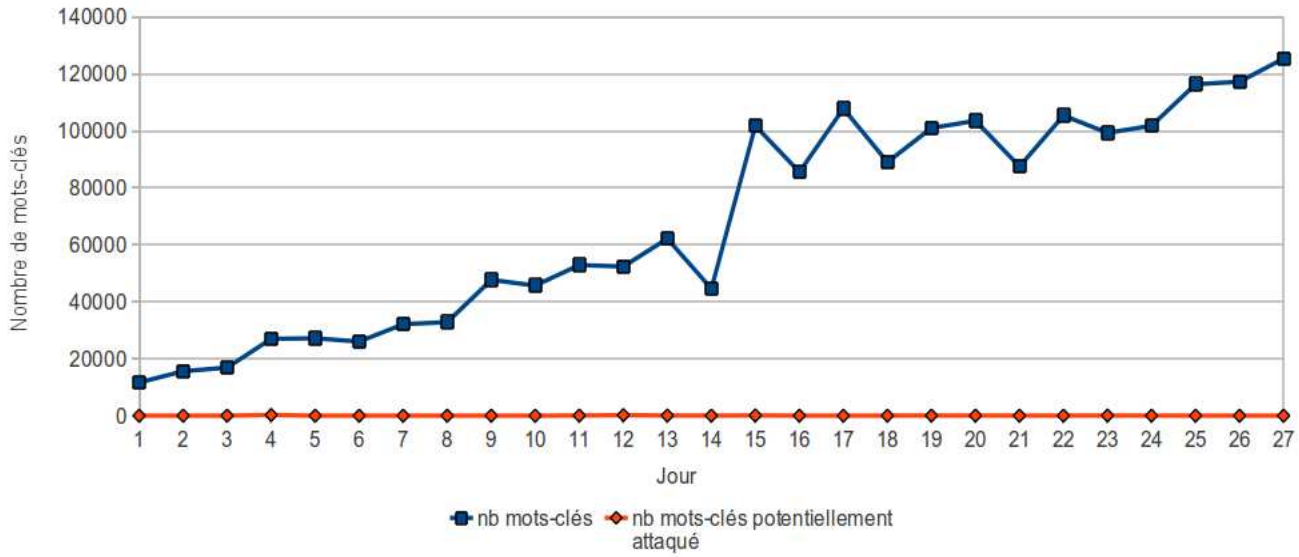

FIGURE 5.2 – Nombre total de mots-clés supervisés et potentiellement attaqués par journée

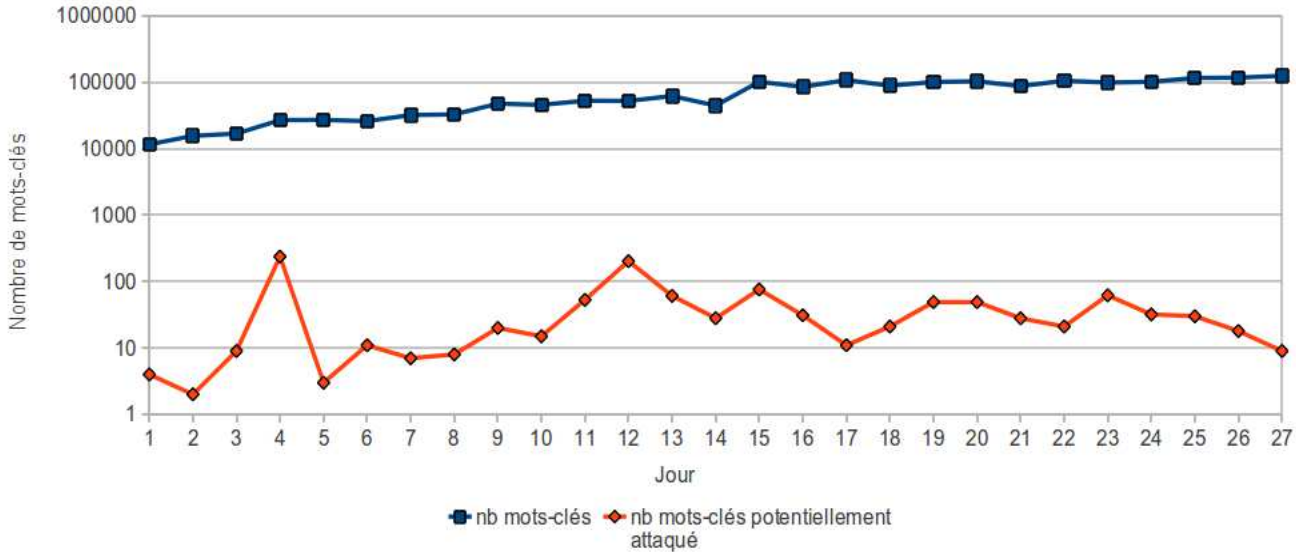

FIGURE 5.3 – Nombre total de mots-clés supervisés et potentiellement attaqués par journée (échelle logarithmique)

5.3.2 Analyse des attaques potentielles

Afin de mieux comprendre les attaques potentielles relevées, nous avons tout d'abord cherché à filtrer les faux positifs possibles (1 pair placé à 29 bit / 30 bit). En prenant exemple sur la journée du 10/03, à l'issue du filtrage, seul le mot-clé avatar est clairement attaqué avec des pairs malveillants partageant 74 bits. Le mot-clé « avatar » est cependant présent présents plusieurs fois dans les données d'Amazon ce qui fait que cette attaque est comptée autant de fois qu'apparait le mot-clé dans les contenus multimédia enregistrés.

```
grep "avatar" XML.2011_3_10_9_12_Rep_Kw.xml  | wc - l
     16      16     368 -
```

Ainsi, sur les 31 mots-clés suspects du 10/03, une seule et même attaque est comptée 16 fois et les 15 autres sont des faux-positifs. Il apparaît que les attaques potentielles relevées pour les autres jours suivent le même schéma avec une attaque sur « avatar » comptée plusieurs fois et quelques faux-positifs dont le nombre varie d'une journée à l'autre selon le placement des pairs. En particulier, l'augmentation du nombre d'attaques potentielles observées les 04/03/2011 et 12/03/2011 203 (figure 5.3) comptant respectivement 203 et 238 attaques s'explique par la présence de faux-positifs sur un mot-clé très populaire et compté de nombreuses fois, en l'occurence les mots-clés « love » et « complete ».

Devant le faible nombre d'attaques mesurées, qui semble contradictoire avec les résultats obtenus dans le chapitre précédent, nous avons souhaité valider les mesures obtenues par notre outil en utilisant l'explorateur (« crawler ») utilisé précédemment. Le 10 mars 2011, nous avons ainsi réalisé une exploration du réseau en parallèle de notre client de mesure puis nous avons étudié les placements de pairs suspects à partir des données du crawler en utilisant la méthode présentée en section 4.3.2. Les résultats présentés ci-dessous montrent qu'il a bien conformité

entre les résultats obtenus par le crawler et notre outil de mesure, à savoir une seule et unique attaque trouvée sur « avatar » à 74 bits. L'explorateur ayant été lui même validé précédemment, nous concluons que les données obtenues par notre outil de mesure ne sont pas faussées.

```
Input: top_kadamazon 07/03/2011
--------------------------------
#java -Xmx1900m crawlerstats.Main -hash ../data/kadamazon_list.txt
crawl_kadamazon.txt 35 avatar C0F70911A9C2E6F6960DDED0D4118244

<C0F70911A9C2E6F696232352F5A279E8, 123.144.160.81, 0, 29762, 8, T> [prefix = 74]
<C0F70911A9C2E6F696232352F5A279E8, 123.145.172.31, 0, 29762, 8, T> [prefix = 74]
<C0F70911A9C2E6F69627D0740C2802BD, 123.144.163.218, 0, 29761, 8, T> [prefix = 74]
<C0F70911A9C2E6F69627D0740C2802BD, 123.144.160.78, 0, 29761, 8, T> [prefix = 74]
<C0F70911A9C2E6F6962DFED0D4118200, 188.165.75.24, 4662, 4672, 8, T> [prefix = 74]
<C0F70911A9C2E6F69630351C55D267D7, 58.17.146.135, 11699, 11699, 8, T> [prefix = 74]
<C0F70911A9C2E6F69630351C55D267D7, 123.144.160.208, 11699, 11699, 8, T> [prefix = 74]
<C0F70911A9C2E6F6963CBCC3B951C1A7, 123.144.163.213, 0, 29763, 8, T> [prefix = 74]
<C0F70911A9C2E6F6963CBCC3B951C1A7, 123.144.167.199, 0, 29763, 8, T> [prefix = 74]

1/2208 of the top-keywords are attacked at at least 35 bits (5486 non-unique)
These 1 top-keywords are attacked by
    9 unique IP addresses (showing 414 unique KADIDs in the whole crawler's data)
    6 subnets /24 (showing 4001 unique KADIDs in the whole crawler's data)
    4 subnets /16 (showing 9733 unique KADIDs in the whole crawler's data)
```

Nous avons également utilisé la base de données de contenus multimédia afin de détecter les attaques trouvées sur les précédentes données d'exploration recueillies le 29/06/2010. Il apparait que les mot-clé d'Amazon permettent bien de détecter les attaques mises en évidence alors (402 mots-clés attaqués) et dont la capture ci-dessous résume les résultats :

```
402/2208 of the top-keywords are attacked at at least 35 bits
These 402 top-keywords are attacked by
    54 unique IP addresses (showing 2600 unique KADIDs in the whole crawler's data)
    53 subnets /24 (showing 2660 unique KADIDs in the whole crawler's data)
    45 subnets /16 (showing 7774 unique KADIDs in the whole crawler's data)
```

5.4 Conclusion

Au terme de ce chapitre, il apparait que les conditions actuelles du réseau et le peu d'attaques recensées durant cette seconde campagne de mesure rendent impossible une meilleure caractérisation des comportements des attaquants. Il apparait en outre que les attaques réalisées sur le réseau KAD sont très changeantes dans le temps puisqu'à six mois d'intervalle, le nombre de mots-clés attaqués a été réduit de plusieurs centaines à un seul. Cela s'explique en partie par le fait que les nombreuses attaques détectées précédemment sont en réalité le fait de peu d'attaquants, en particulier deux motifs d'attaques différents pouvaient être mis en évidence dans le chapitre précédent.

Chapitre 6

Conclusion et travaux à venir

Dans le cadre du projet GIS 3SGS ACDAP2P, nous étudions la possibilité d'utiliser une approche collaborative pour la détection d'attaques sur les réseaux pair à pair. Dans ce contexte, le premier travail a consisté à effectuer l'état de l'art des réseaux pair à pair et de leur sécurité en terme de failles et solutions collaboratives pour la détection. Le présent délivrable a présenté nos premières contributions allant dans ce sens à savoir l'identification des vulnérabilités de la DHT de BitTorrent ainsi que la détection des comportements malveillants dans KAD. Chacune de ces deux contributions a été validée par une publication.

Nous avons tout d'abord rappelé le fonctionnement des services d'indexation réalisés au dessus de la DHT de KAD et les attaques réalisées sur celles-ci. Nous avons proposé une taxonomie des attaques en distinguant notamment deux grandes familles d'attaques : les attaques internes basées sur l'insertion de noeuds malveillants dans le réseau et les attaques externes basées sur l'envoi de messages.

Dans le présent rapport, nous nous sommes focalisés sur l'étude des attaques internes en montrant tout d'abord la vulnérabilité du nouveau service d'indexation distribué de BitTorrent à celles-ci [TCCF11] [1]. Nous avons conçu et mis en oeuvre une architecture d'expérimentation permettant de réaliser des attaques sur le réseau réel, puis, nous avons montré l'applicabilité de mesures de protection précédemment étudiées dans KAD et qui permettraient de protéger ce réseau. Nous avons ensuite proposé une approche permettant de détecter, de manière centralisée, des attaques internes localisées sur le réseau KAD [CHC+11] [2]. Nous avons pour cela réalisé un explorateur permettant de découvrir le réseau et nous avons mis en évidence de nombreuses attaques en étudiant les distances entre les pairs et entre les pairs et les contenus indexés. Enfin, nous avons souhaité caractériser plus précisément les comportements des attaquants. Nous avons pour cela procédé à des mesures régulières sur le réseau et guidées par une base de données de contenus multimédia constamment mise à jour. Malheureusement, l'évolution des attaques sur le réseau KAD n'a pas permis la collecte de données significatives sur les attaquants mais nous a permis de constater la disparité temporelles des attaques affectant le réseau.

Le prochain délivrable porte sur la proposition d'une solution collaborative permettant de détecter les comportements malveillants dans les réseaux P2P. Etant donné le nouveau contexte du réseau KAD, notre étude ne portera pas sur les attaques internes comme prévu initialement

1. http://hal.inria.fr/inria-00577043/PDF/BitTorrent_DHT_security_assessment_ntms11.pdf
2. http://hal.inria.fr/inria-00596677/PDF/SARSSI11-Detection_Attaques_KAD-Cholez.pdf

car celles-ci sont actuellement limitées, mais sur les attaques externes engendrant l'immense pollution constatée quotidiennement par les utilisateurs du réseau. Nous proposerons ainsi, dans le prochain délivrable, une métrique capable de détecter précisément les contenus pollués puis nous terminerons le projet en proposant une approche collaborative basée sur cette métrique et capable de limiter la diffusion de la pollution.

Bibliographie

[CCF09] Thibault Cholez, Isabelle Chrisment, and Olivier Festor. Evaluation of Sybil Attacks Protection Schemes in KAD. In *3rd International Conference on Autonomous Infrastructure, Management and Security - AIMS 2009*, volume 5637 of *Lecture Notes in Computer Science*, pages 70–82, Enschede Pays-Bas, 2009. University of Twente, Springer.

[CCF10a] Thibault Cholez, Isabelle Chrisment, and Olivier Festor. Efficient DHT attack mitigation through peers' ID distribution. In *Seventh International Workshop on Hot Topics in Peer-to-Peer Systems - HotP2P 2010*, Atlanta États-Unis, 04 2010. IEEE International Parallel & Distributed Processing Symposium.

[CCF10b] Thibault Cholez, Isabelle Chrisment, and Olivier Festor. Monitoring and Controlling Content Access in KAD. In *International Conference on Communications - ICC 2010*, Capetown Afrique Du Sud, 05 2010. IEEE.

[CHC+11] Thibault Cholez, Christopher Hénard, Isabelle Chrisment, Olivier Festor, Guillaume Doyen, and Rida Khatoun. Détection de pairs suspects dans le réseau pair à pair KAD. In *SAR-SSI 2011 : 6ème Conf. sur la Sécurité des Architectures Réseaux et Systèmes d'Information*, La Rochelle, France, May 2011. IEEE. Financement GIS - 3SGS - Projet ACDAP2P.

[Cho11] Thibault Cholez. *Supervision des réseaux pair à pair structurés appliquée à la sécurité des contenus*. These, Université Henri Poincaré - Nancy I, June 2011.

[Coh03] Bram Cohen. Incentives build robustness in bittorrent. Technical report, bittorrent.org, 2003.

[CW07] Scott A. Crosby and Dan S. Wallach. An analysis of bittorrent's two kademlia-based dhts, 2007.

[Dou02] John R. Douceur. The sybil attack. In *IPTPS '01 : Revised Papers from the First International Workshop on Peer-to-Peer Systems*, pages 251–260, London, UK, 2002. Springer-Verlag.

[Ipo09] Ipoque. Internet study 2008/2009. `http://www.ipoque.com/resources/internet-studies/internet-study-2008_2009`, 2009.

[JDH10] Oliver Jetter, Jochen Dinger, and Hannes Hartenstein. Quantitative analysis of the sybil attack and effective sybil resistance in peer-to-peer systems. In *In IEEE ICC 2010 proceedings*, 2010.

[JOK09] Raúl Jiménez, Flutra Osmani, and Björn Knutsson. Connectivity properties of mainline bittorrent dht nodes. In Henning Schulzrinne, Karl Aberer, and Anwitaman Datta, editors, *Peer-to-Peer Computing*, pages 262–270. IEEE, 2009.

[KCW10] Jie Kong, Wandong Cai, and Lei Wang. The evaluation of index poisoning in bittorrent. *Communication Software and Networks, International Conference on*, 0 :382–386, 2010.

[KCWZ10] Jie Kong, Wandong Cai, Lei Wang, and Qiushi Zhao. A study of pollution on bittorrent. In *Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on*, volume 3, pages 118 –122, 26-28 2010.

[KLR09] Michael Kohnen, Mike Leske, and Erwin P. Rathgeb. Conducting and optimizing eclipse attacks in the KAD peer-to-peer network. In *NETWORKING '09 : Proceedings of the 8th International IFIP-TC 6 Networking Conference*, pages 104–116, Berlin, Heidelberg, 2009. Springer-Verlag.

[LBLF+10] Stevens Le-Blond, Arnaud Legout, Fabrice Le Fessant, Walid Dabbous, and Mohamed Ali Kâafar. Spying the world from your laptop – identifying and profiling content providers and big downloaders in bittorrent. *CoRR*, abs/1004.0930, 2010.

[LMSW10] Thomas Locher, David Mysicka, Stefan Schmid, and Roger Wattenhofer. Poisoning the Kad Network. In *11th International Conference on Distributed Computing and Networking (ICDCN), Kolkata, India*, January 2010.

[LNR06] J. Liang, N. Naoumov, and K. W. Ross. The Index Poisoning Attack in P2P File Sharing Systems. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–12. IEEE, 2006.

[Loe08] Andrew Loewenstern. DHT protocol. `http://bittorrent.org/beps/bep_0005.html`, 2008.

[LSM06] Brian Neil Levine, Clay Shields, and N. Boris Margolin. A Survey of Solutions to the Sybil Attack. Tech report 2006-052, University of Massachusetts Amherst, Amherst, MA, October 2006.

[MM02] Petar Maymounkov and David Mazières. Kademlia : A peer-to-peer information system based on the xor metric. In *IPTPS '01 : Revised Papers from the First International Workshop on Peer-to-Peer Systems*, pages 53–65, London, UK, 2002. Springer-Verlag.

[MRGS09] Ghulam Memon, Reza Rejaie, Yang Guo, and Daniel Stutzbach. Large-scale monitoring of DHT traffic. In *International Workshop on Peer-to-Peer Systems (IPTPS)*, Boston, MA, April 2009.

[NR06] Naoum Naoumov and Keith Ross. Exploiting p2p systems for ddos attacks. In *InfoScale '06 : Proceedings of the 1st international conference on Scalable information systems*, page 47, New York, NY, USA, 2006. ACM.

[PKK08] Michael Piatek, Tadayoshi Kohno, and Arvind Krishnamurthy. Challenges and directions for monitoring p2p file sharing networks-or : why my printer received a dmca takedown notice. In *HOTSEC'08 : Proceedings of the 3rd conference on Hot topics in security*, pages 1–7, Berkeley, CA, USA, 2008. USENIX Association.

[SENB07a] Moritz Steiner, Taoufik En-Najjary, and Ernst W. Biersack. Exploiting kad : possible uses and misuses. *SIGCOMM Comput. Commun. Rev.*, 37(5) :65–70, 2007.

[SENB07b] Moritz Steiner, Taoufik En-Najjary, and Ernst W Biersack. A global view of kad. In *IMC 2007, ACM SIGCOMM Internet Measurement Conference, October 23-26, 2007, San Diego, USA*, 10 2007.

[SPR09] Georgos Siganos, Josep Pujol, and Pablo Rodriguez. Monitoring the bittorrent monitors : A bird's eye view. pages 175–184. 2009.

[TCCF11] Juan Pablo Timpanaro, Thibault Cholez, Isabelle Chrisment, and Olivier Festor. BitTorrent's Mainline DHT Security Assessment. In *4th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, Paris France, 02 2011. IEEE. Projet GIS 3SGS ACDAP2P (Approche collaborative pour la détection d'attaques dans les réseaux pair à pair).

[UPvS09] Guido Urdaneta, Guillaume Pierre, and Maarten van Steen. A survey of DHT security techniques. *ACM Computing Surveys*, 2009. `http://www.globule.org/publi/SDST_acmcs2009.html`, to appear.

[UPvS11] Guido Urdaneta, Guillaume Pierre, and Maarten van Steen. A survey of DHT security techniques. *ACM Computing Surveys*, 43(2), June 2011. `http://www.globule.org/publi/SDST_acmcs2009.html`.

[WH] Scott Wolchok and J. Alex Halderman. Crawling bittorrent dhts for fun and profit.

[WTCT+08] Peng Wang, James Tyra, Eric Chan-Tin, Tyson Malchow, Denis Foo Kune, Nicholas Hopper, and Yongdae Kim. Attacking the kad network. In *SecureComm '08 : Proceedings of the 4th international conference on Security and privacy in communication netowrks*, pages 1–10, New York, NY, USA, 2008. ACM.

[YFX+09] Jie Yu, Chengfang Fang, Jia Xu, Ee-Chien Chang, and Zhoujun Li. Id repetition in kad. In *Peer-to-Peer Computing'09*, pages 111–120, Atlanta États-Unis, 09 2009. IEEE.